**Name:** Muhammad Mehdi Ali **Roll No:** BSEF18M015

**GitHub Tutorial:-**

**Steps:-**

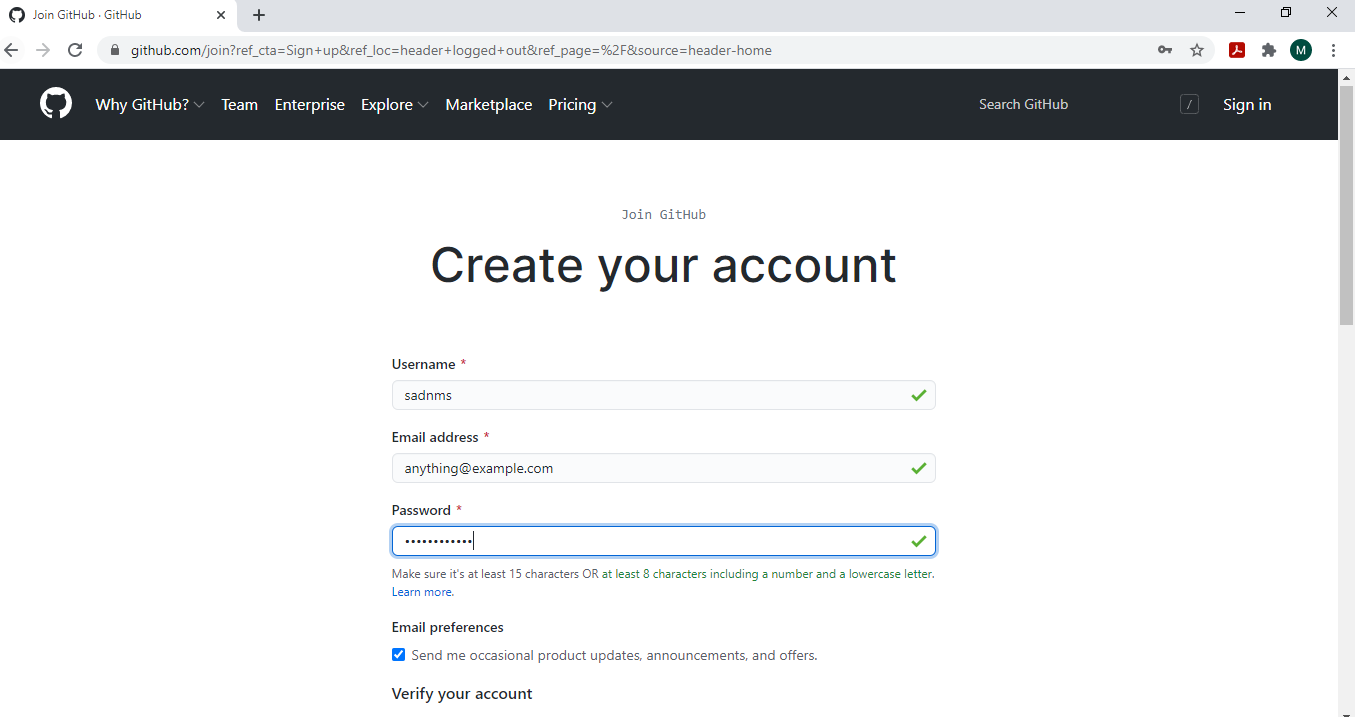
**Getting Started:-**

1. Go to github.com

A screenshot of a computer screen

Description generated with high confidence

1. Click on sign up and make an account



1. Then Sign in to your account.

Graphical user interface, application

Description generated with very high confidence

1. Your account will be shown

Graphical user interface, application

Description generated with very high confidence

1. Click on + button to add a repo

Graphical user interface, application

Description generated with very high confidence

**Making a repository:-**

1. Click on New Repository

Graphical user interface, application

Description generated with very high confidence

1. Give details of the repository

Graphical user interface, text, application, email

Description generated with very high confidence

1. Click on Create Repository
2. Copy The link of your repository we will use it later

Graphical user interface, text, application, email

Description generated with very high confidence

1. Create a directory and open git bash in it

A picture containing text

Description generated with very high confidence

**Manage repository locally and git add, commit and push commands:-**

1. Clone repository locally using git clone command and url copied at step 9

Text

Description generated with very high confidence

12)Create a file and add data to it

Graphical user interface, text, application

Description generated with high confidence

1. Now push this txt file to main repo using git add, commit and push commands

Text

Description generated with very high confidence

1. Changes will be applied at main repository at github.com too.

A screenshot of a computer

Description generated with high confidence

**Git Pull:-**

1. Edit the file locally

Graphical user interface, text, application

Description generated with high confidence

1. Now update the main repository by using git add, commit and push commands

Text

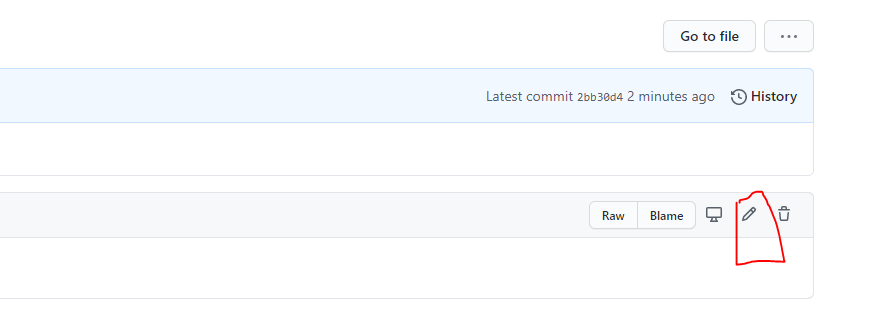
Description generated with very high confidence

1. Main Repository will be updated

A screenshot of a computer

Description generated with high confidence

1. Now to edit from online github editor click on edit button at top right

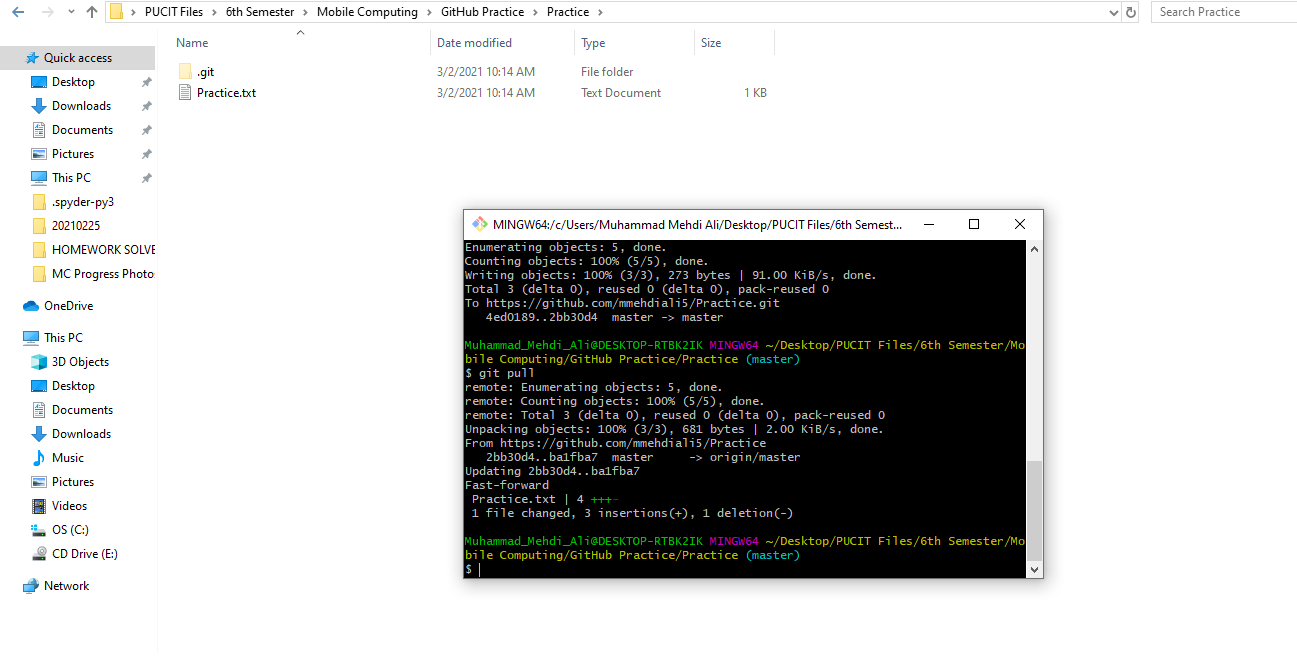


18)Edit the document there

Graphical user interface, text, application, email

Description generated with very high confidence

19)Now to update local repo we will use git pull command



20) Local repo will be updated

Graphical user interface, text, application

Description generated with high confidence

**MERGE CONFLICTS:-**

Suppose that two people change same line of the file one person change it locally and other on main branch. In this case if we want to push changes to main repository we will get error of merge conflicts which must be resolved before pushing

1. Open the file in text Editor and change a line in it

Text

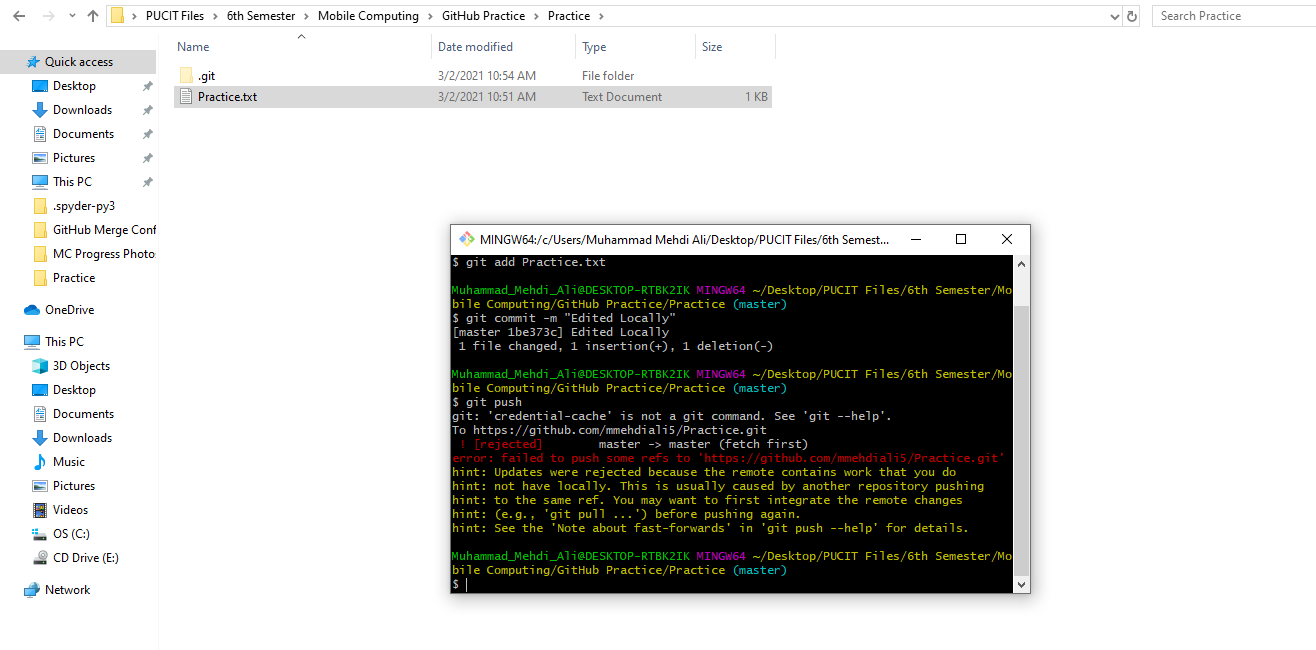
Description generated with very high confidence

1. Now change the line at main repo and commit

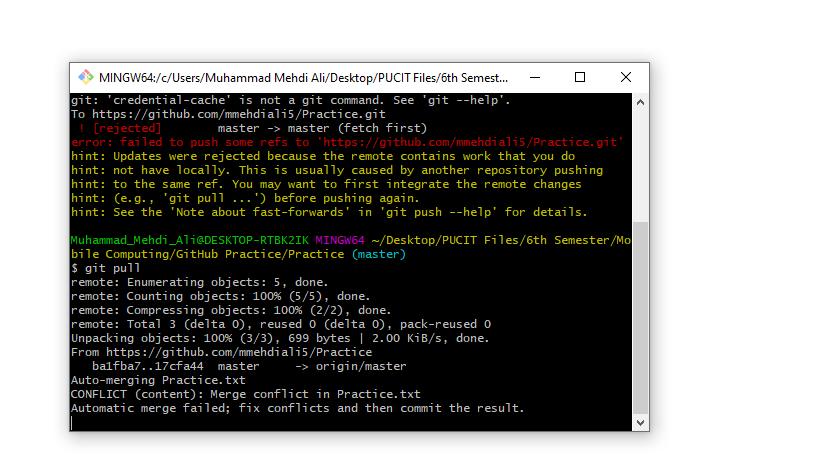
Graphical user interface, text, application, email

Description generated with high confidence

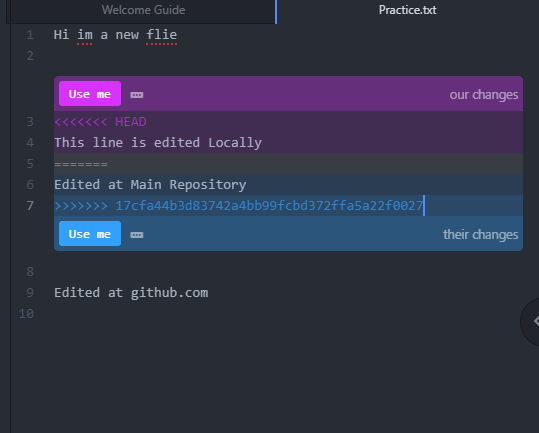
1. Now when we try to push from local repository we get an conflict error message



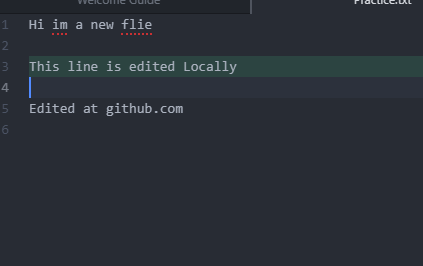
1. To resolve this use git pull to pull the file from main repository



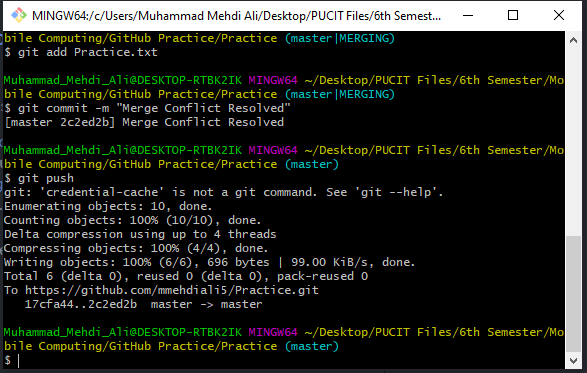
1. When you open the file you will see buttons use me click on the line which you want to retain



1. Choose one of the lines



1. Now we can push successfully and merge conflict is resolved



1. Main repo will also be updated

Graphical user interface, application, Word

Description generated with very high confidence

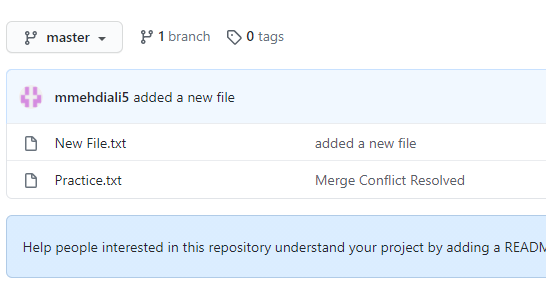
**Git Remove**

1. Let us add a file in our repository

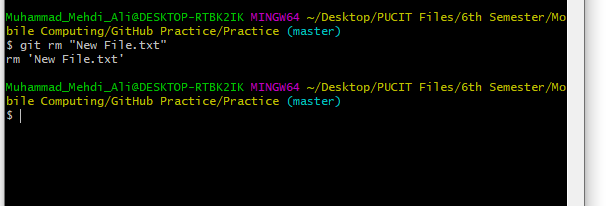
Graphical user interface, text, application

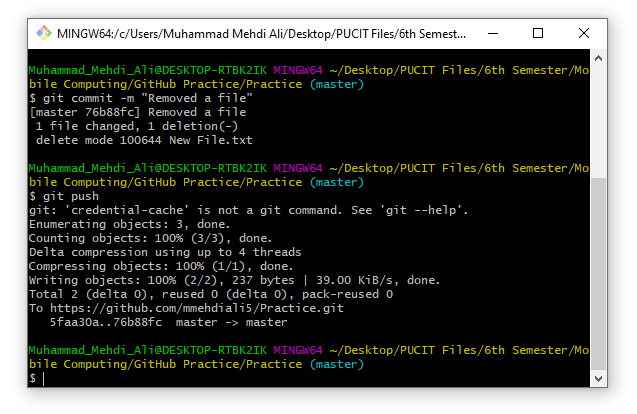
Description generated with very high confidence

1. Now we check it on our main repository

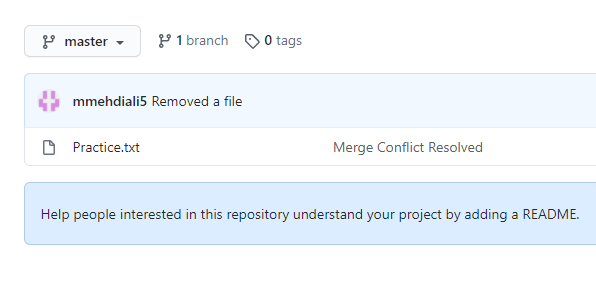


1. Now we remove it using git remove and commit changes and push to main repo





1. Now, we verify it on our main repo



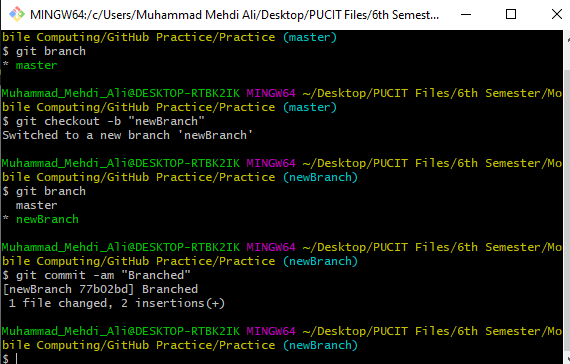
**Git Branch**

1. Let us create a branch in our project so that we can make changes in it separately and marge when required. We use git checkout to navigate among branches. Now we add some text to it.

Graphical user interface, application, Word

Description generated with very high confidence

1. Now we commit it.



1. When we checkout to master we can see that master branch is at was before branching

Graphical user interface, application

Description generated with very high confidence

1. Now we merge it with master and delete the branch. We can see that now master have branch text also

Graphical user interface, application

Description generated with very high confidence

**ANDROID STUDIO**

**First Program:-**

1. First we open our android studio and start a new project.
2. Then we select empty project option
3. After loading we go to our activity\_main.xml
4. We add a text “Application Submission” and adjust the margins

Graphical user interface, application

Description generated with very high confidence

1. Now we add some input text of Name, Address and Email

Graphical user interface, application

Description generated with very high confidence

1. Now we add some checkboxes and submit button

Graphical user interface, application

Description generated with very high confidence

**Layouts:-**

1. In this layout elements are sequentially inserted

Graphical user interface, application

Description generated with very high confidence

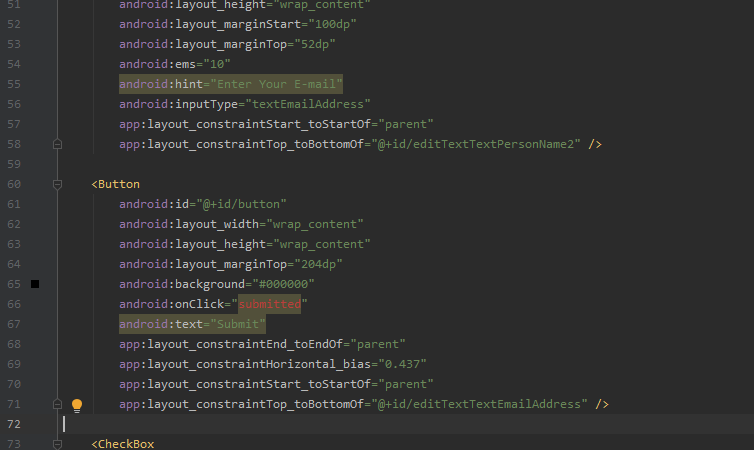
A picture containing text

Description generated with very high confidence

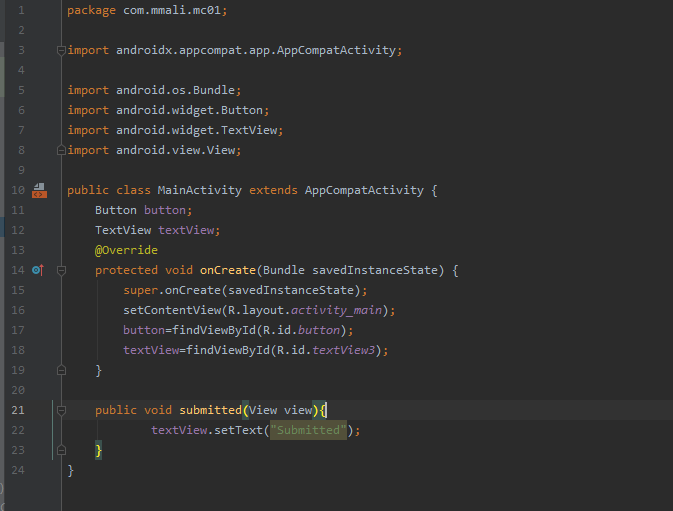


**OnClick:-**

1. Now we will assign an onclick function to our submit function
2. Go to code and add onClick in button tag and write name of the function



1. Now to remove the error we will define the function in main



1. Output on android device

Graphical user interface, application

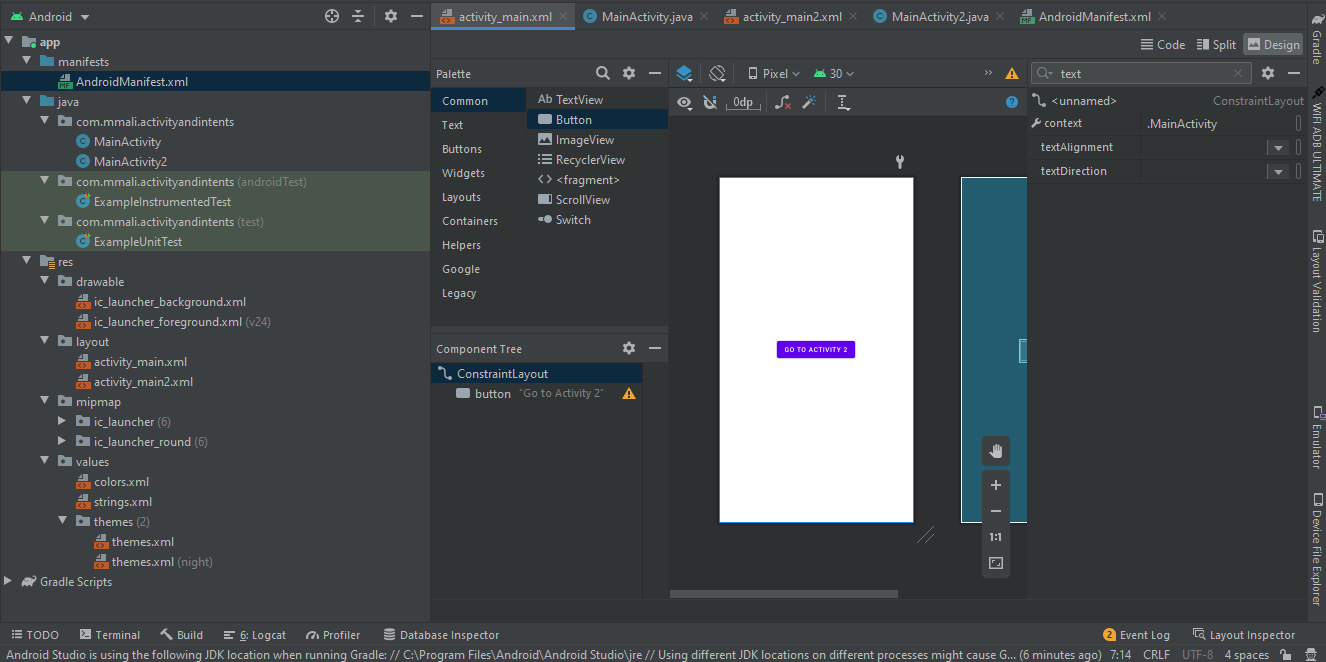
Description generated with high confidence

Graphical user interface, application

Description generated with high confidence

**Activiity and Intend**

1. We first create a main activity with a button in it having text Go to avtivity 2.



1. Now we create an empty activity

Graphical user interface, application

Description generated with very high confidence

1. Now we add 2 buttons here and give them onClick functions

Text

Description generated with very high confidenceGraphical user interface, application

Description generated with very high confidence

1. Now we apply onClick to go to activity 2 from main activity

Text

Description generated with very high confidence

1. Now we will see the output on android phone

Graphical user interface, application, Teams

Description generated with very high confidenceGraphical user interface, application, Teams

Description generated with very high confidence

Graphical user interface, application

Description generated with very high confidence

**ACTIVITY LIFECYCLE**

1. There are three activities performed in application:-

* OnCreate()
* OnStart()
* OnResume()
* OnPause()
* OnStop()
* OnDestroy()

1. When an activity destroys it’s data is also destroyed.
2. For example let us create a counter which keeps count of clicks on button

Text

Description generated with very high confidence

Graphical user interface, application

Description generated with high confidence

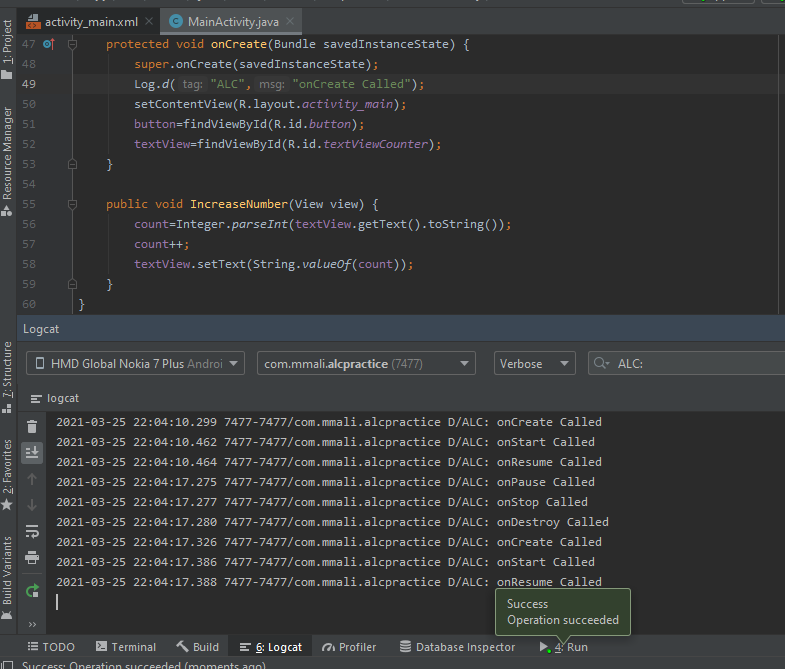
4)Now when we try to increase the count and rotate it we see that it resets to zero its because new activity is started and previous activity is destroyed.

Graphical user interface, application

Description generated with high confidenceGraphical user interface, text, application

Description generated with high confidence

5)Now we log the activities to see what is happening



6)Now to solve this issue we will save the previous state of activity

Text

Description generated with very high confidence

1. Now if we increase count and rotate screen the previous activity state is preserved and count is same as before

Graphical user interface, text, application

Description generated with high confidence

Graphical user interface, text, application

Description generated with high confidence

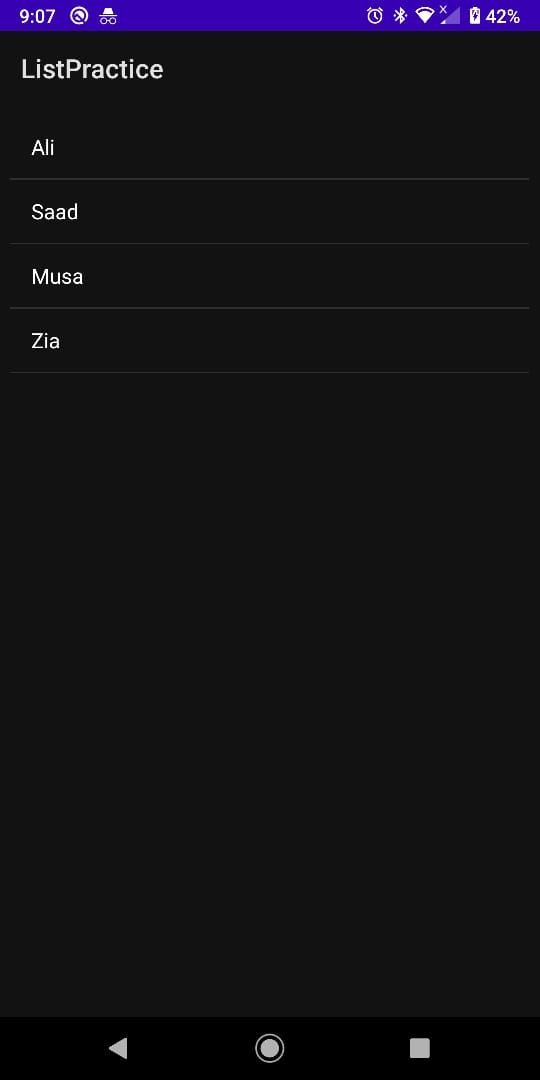
**ADAPTERS**

1)Adapters convert data from a data source in UI Components eg. ListView, GridView etc

2)Let Us make a code for a listView.

Text

Description generated with very high confidence



3)Now we do this procedure for ArrayList

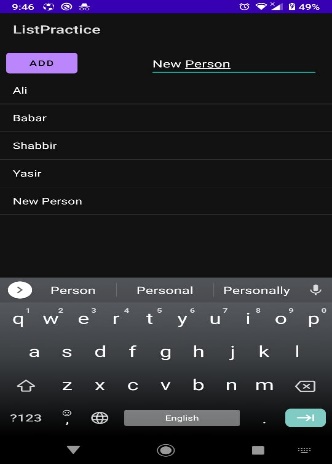
Text

Description generated with very high confidenceText

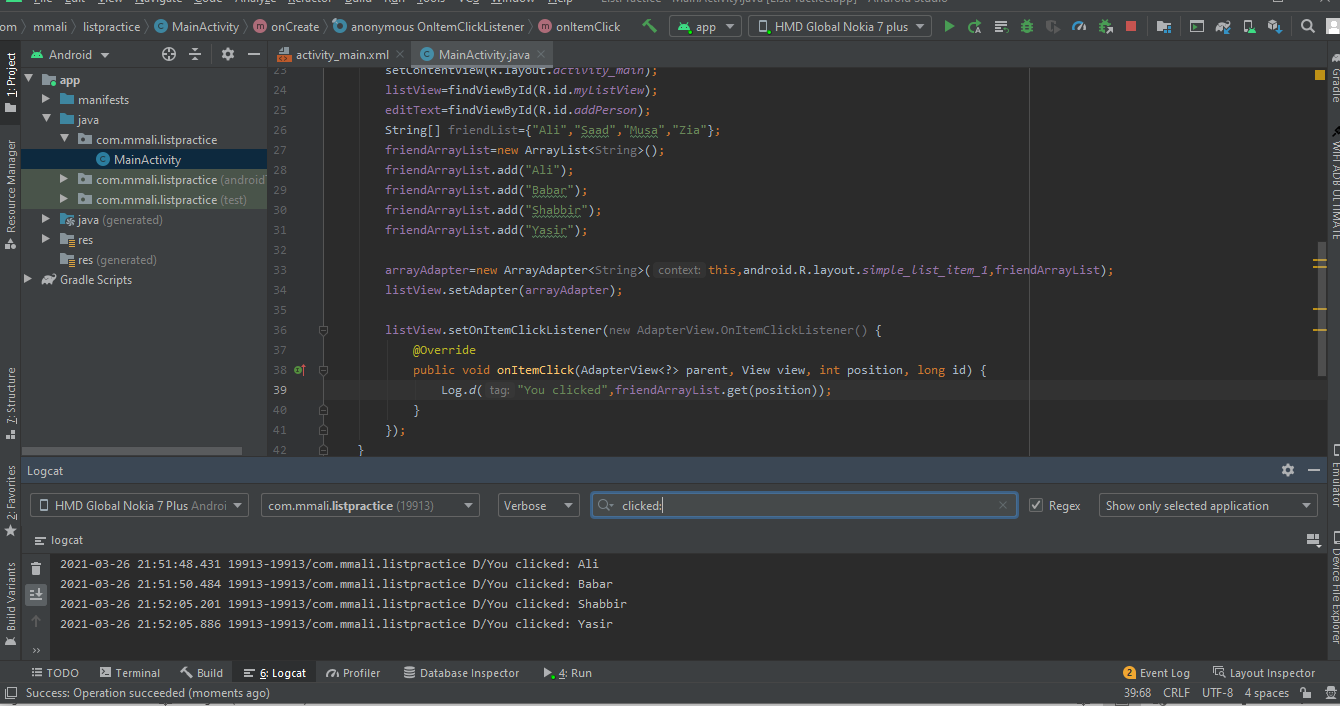
Description generated with high confidence

4)Now we add record taking input from user

Text

Description generated with very high confidence

5)Now we try to add some clickEventListner to list Items



6)Now we update list using notifyDataSetChanged so that whenever our list is changed the listview refresh itself

Text

Description generated with very high confidence