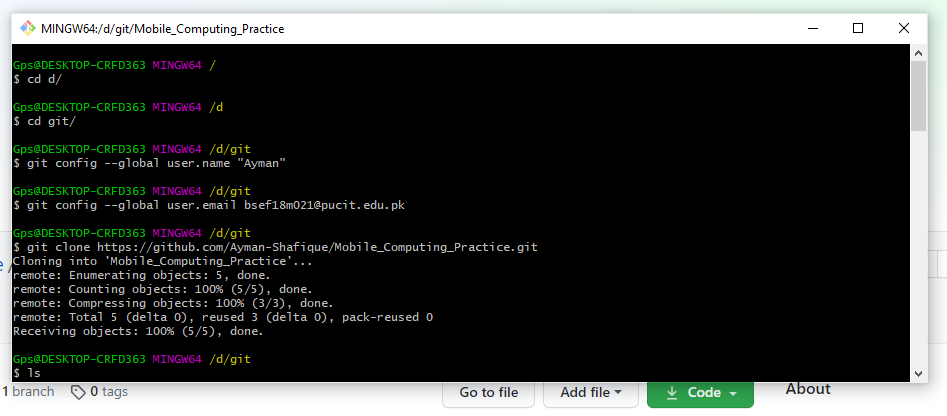
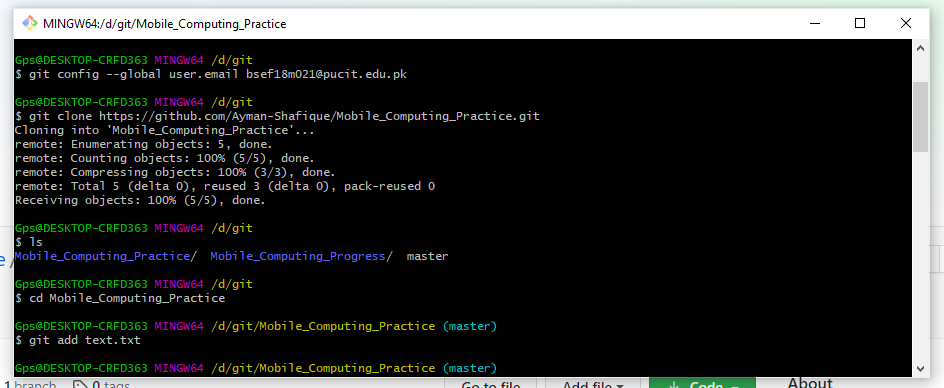
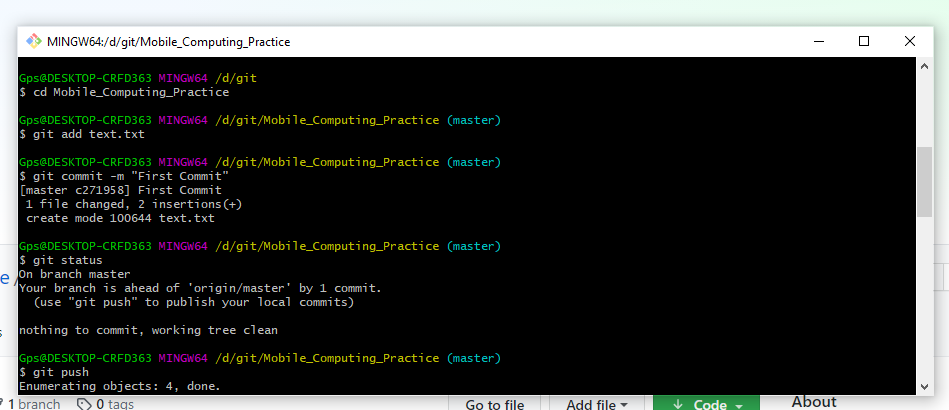
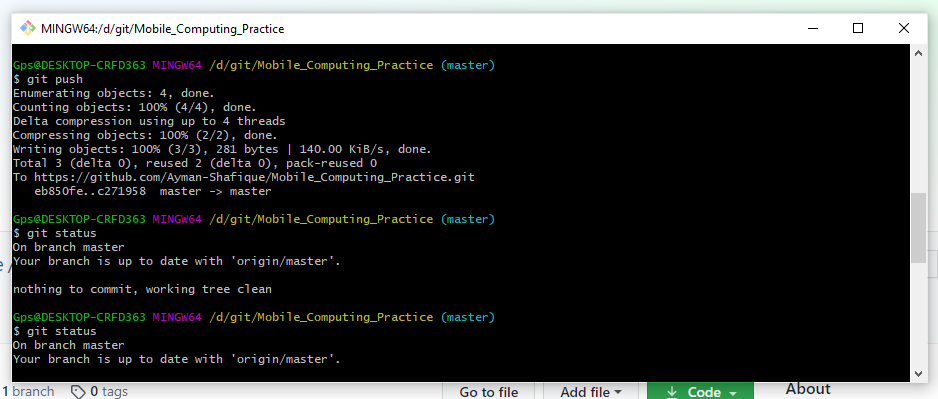
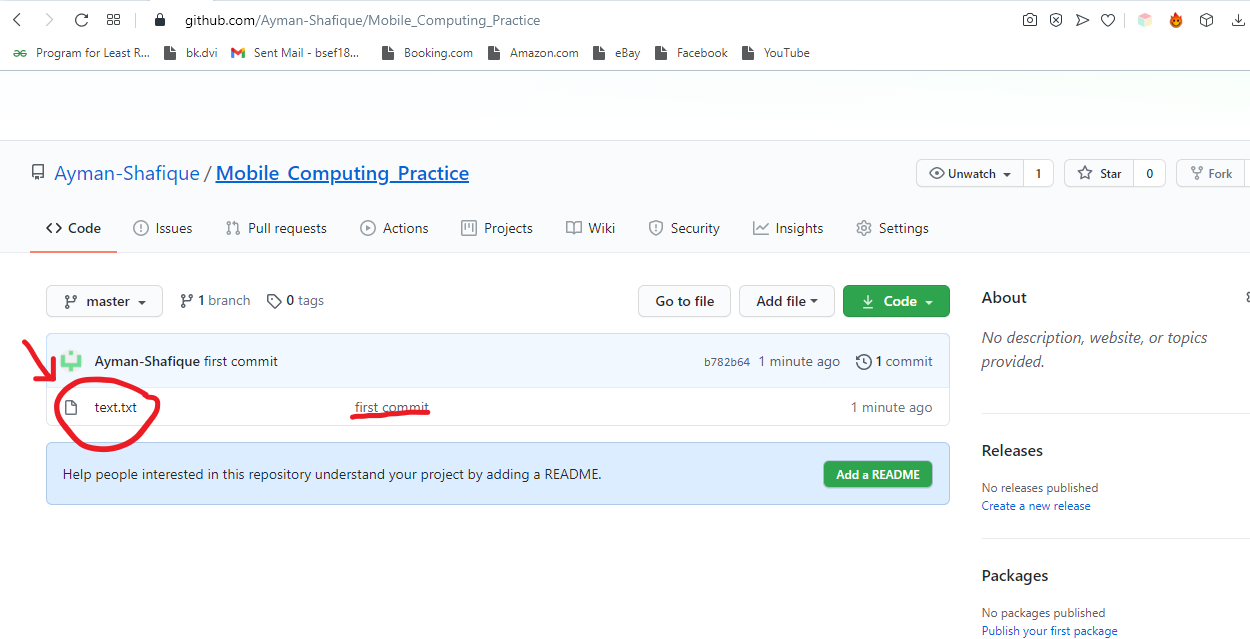
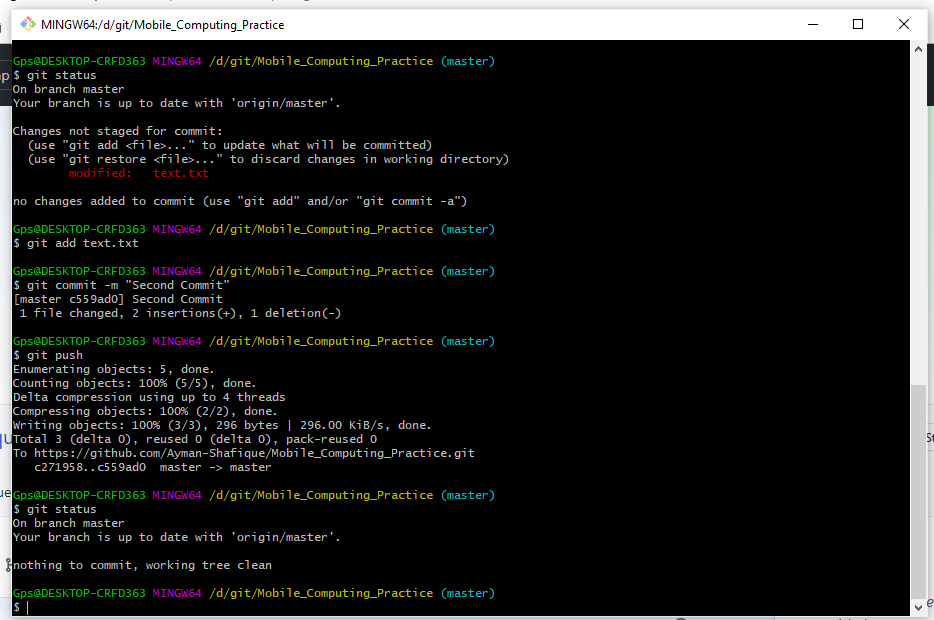
**----------------------------- Lecture # 02 -----------------------------**

**VCS-git**

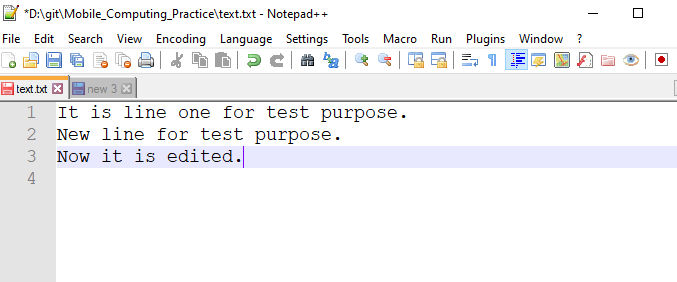
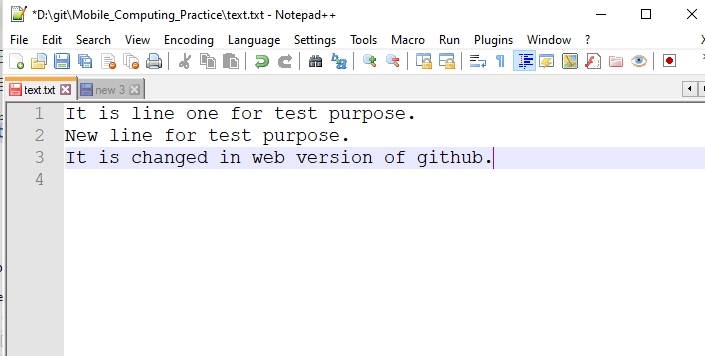
* **Git clone**
  + **git clone** is a **Git** command line utility which is used to target an existing repository and create a **clone**, or copy of the target repository.
  + **Command:**

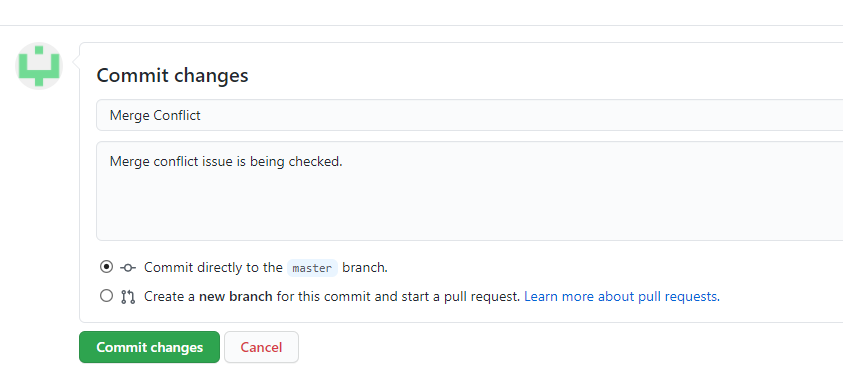
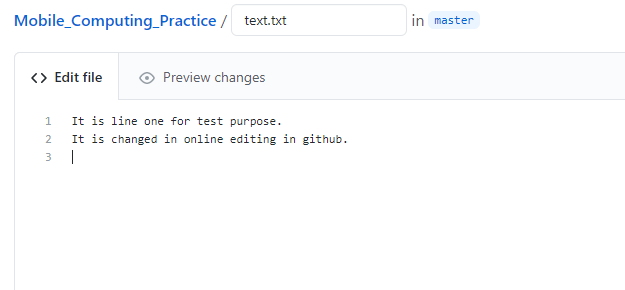
**git clone https://github.com/Ayman-Shafique/Mobile\_Computing\_Practice.git** 

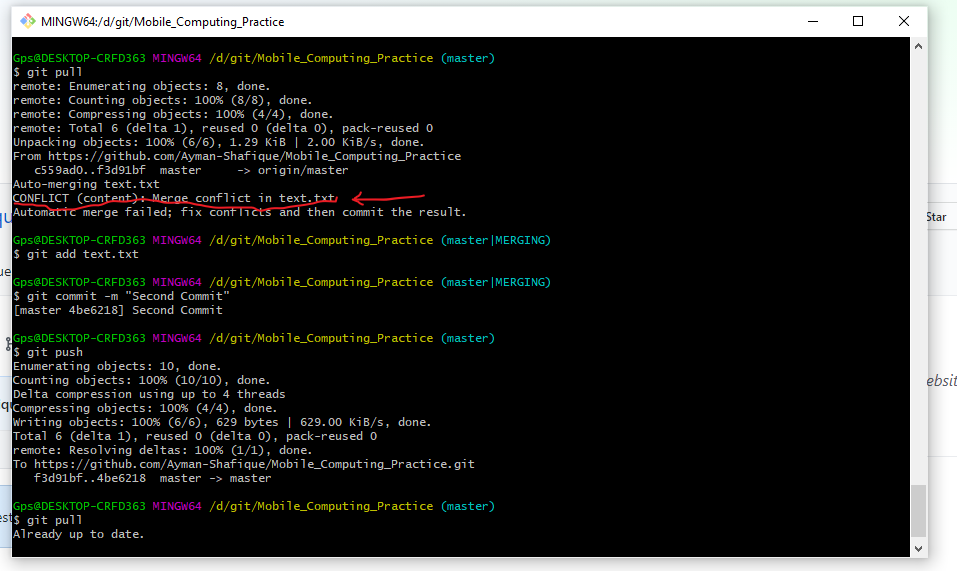
* **Git add**
  + The **git add** command adds a change in the working directory to the staging area.
  + **cd Mobile\_Computing\_Practice**
  + **Command:** git add text.txt
* **Git commit**
  + The **git commit** command captures a snapshot of the project's currently staged changes.
  + **Command:** git commit -m "First Commit"
* **Git push**
  + The **git push** command is used to upload local repository content to a remote repository. **Pushing** is how you transfer commits from your local repository to a remote repository.
  + **Command:** git push
* **Git status**
  + The **git status** command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by **Git**.
  + **Command:** git status

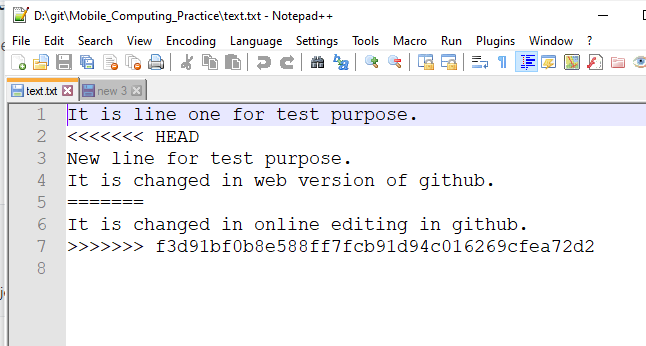
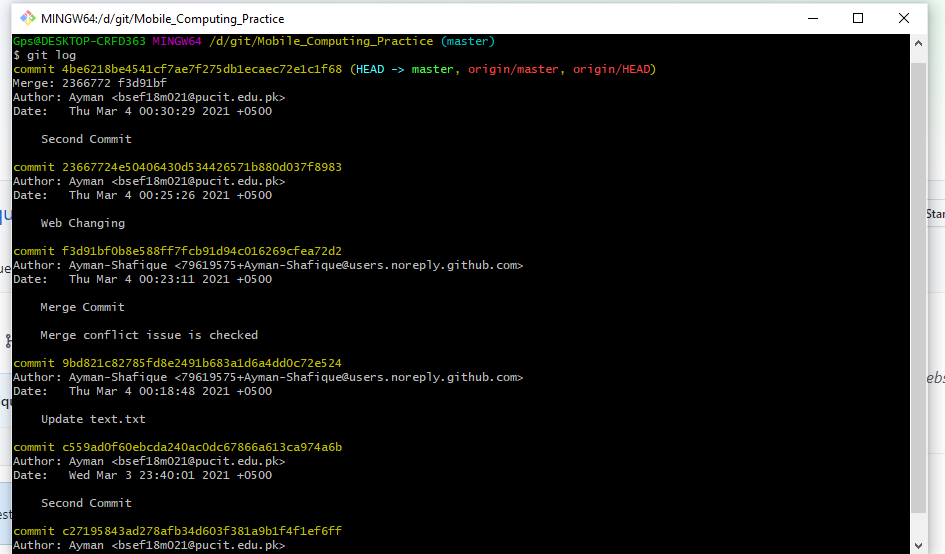
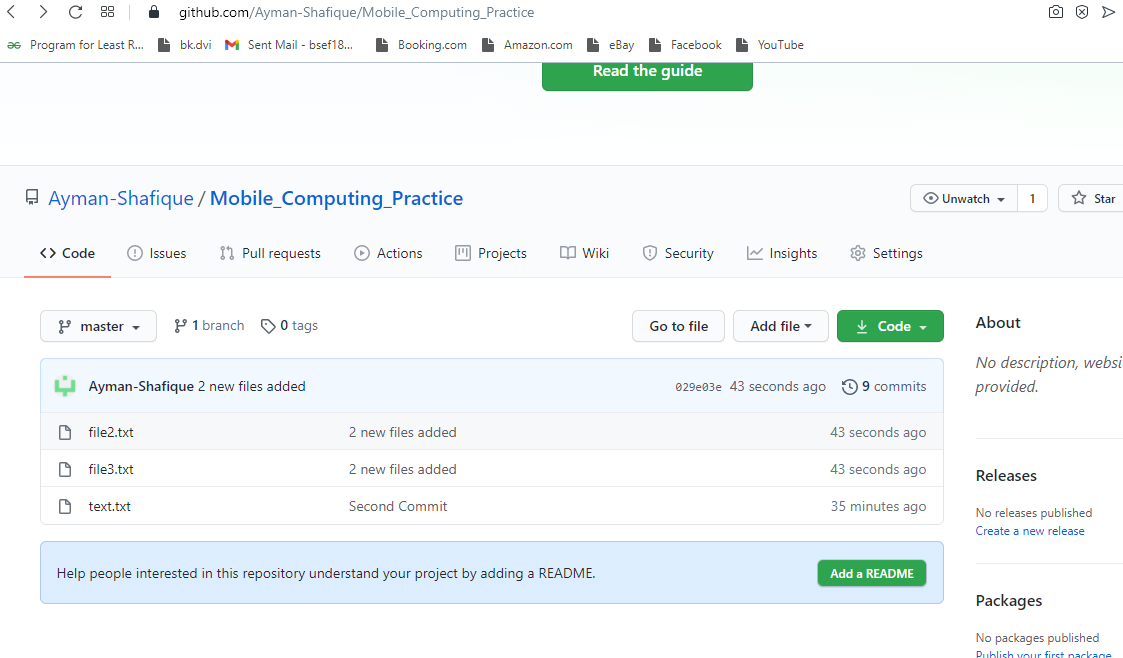
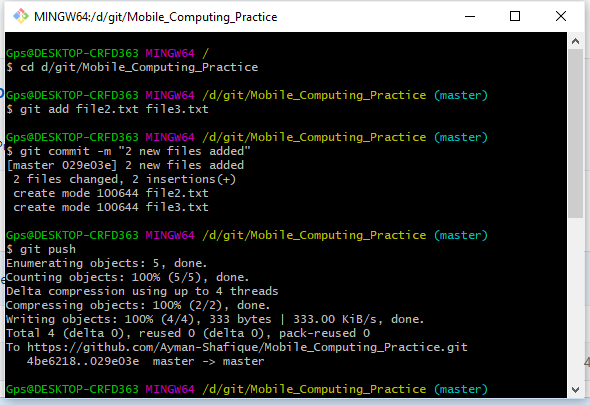
**----------------------------- Lecture # 03 -----------------------------**

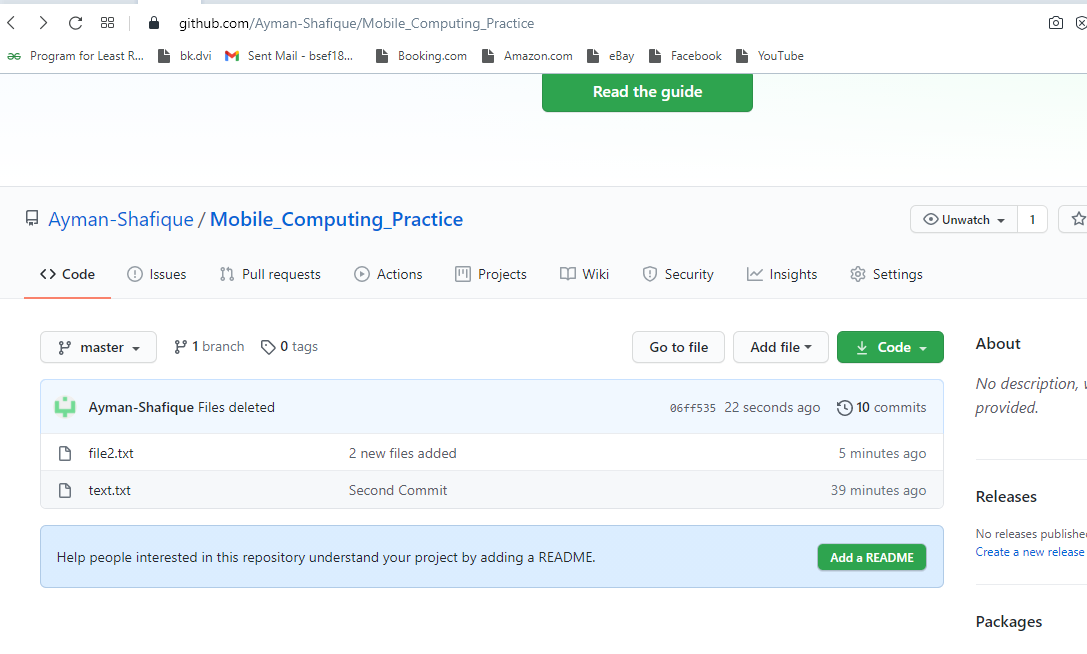
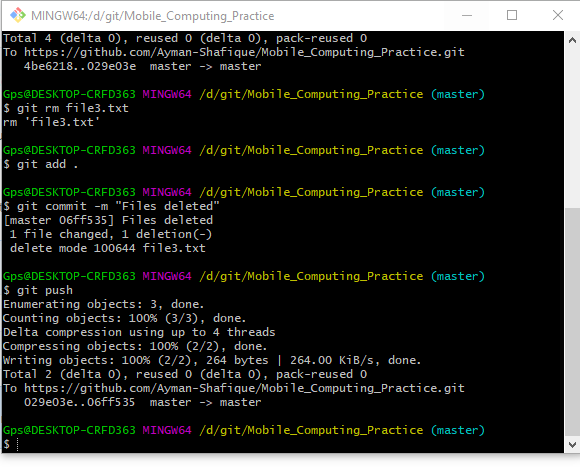
**Merge Conflicts**

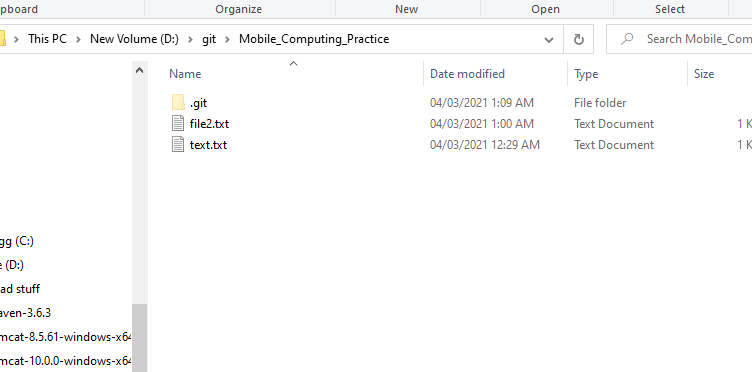
* **Text.txt file before editing:**
* **Editing in Web version of GitHub:**
* **Online Editing:**

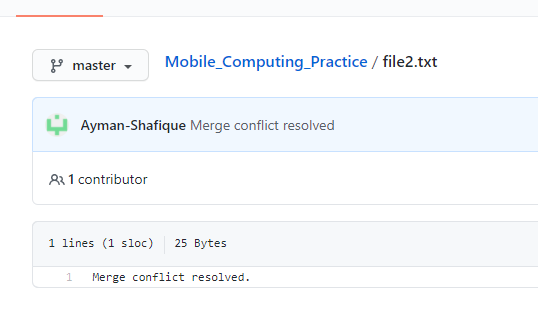
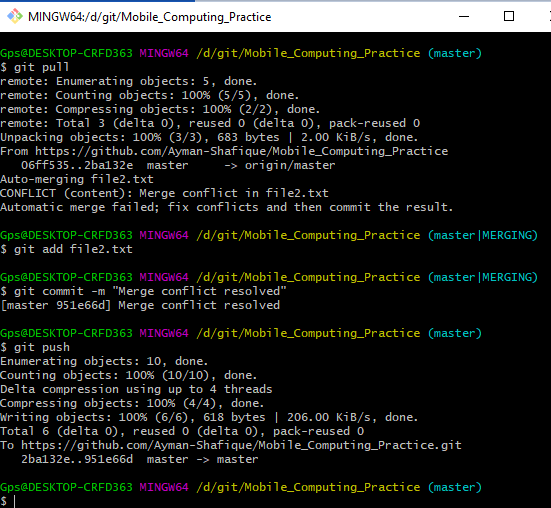
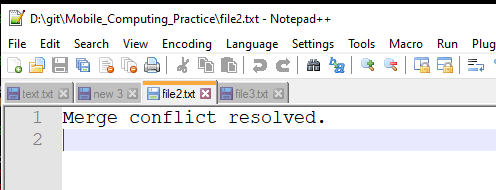
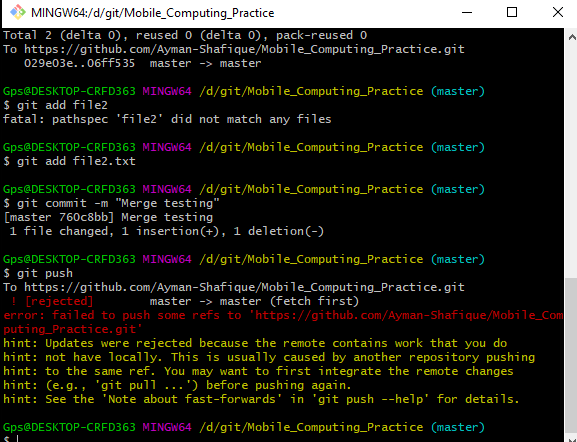
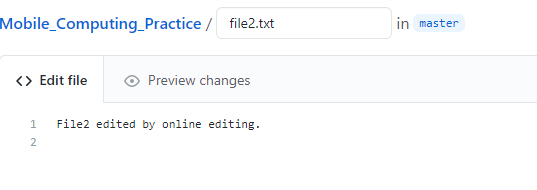
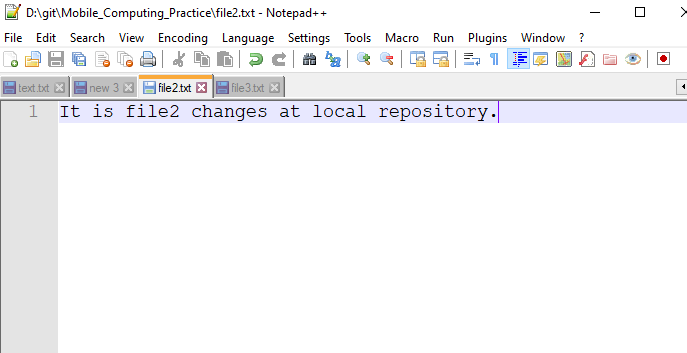


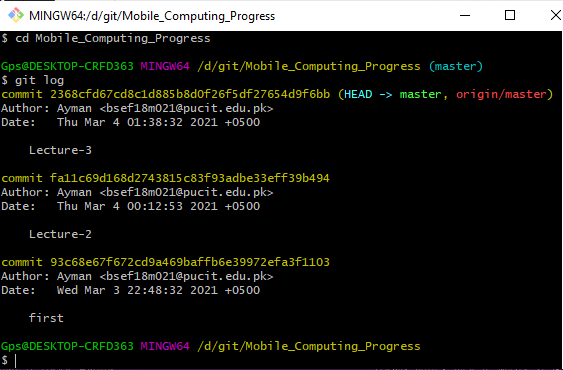
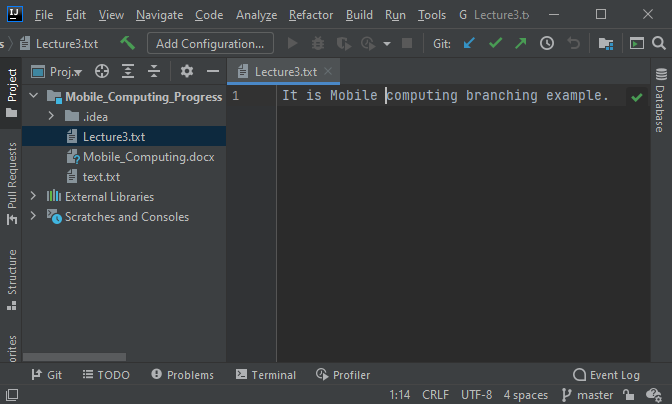
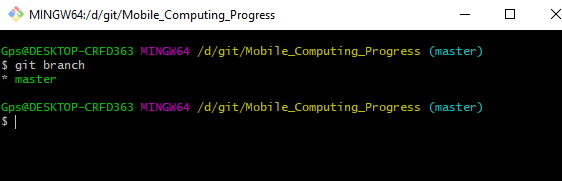
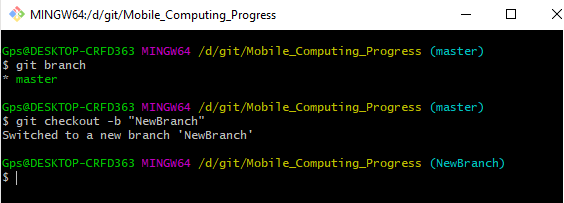
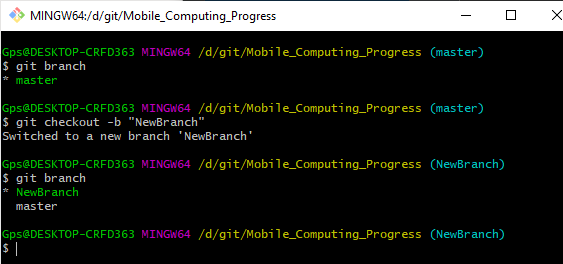
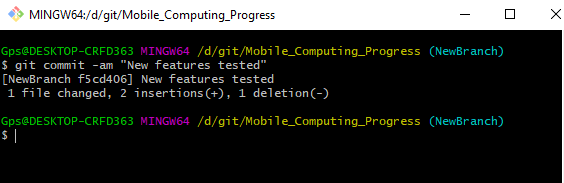
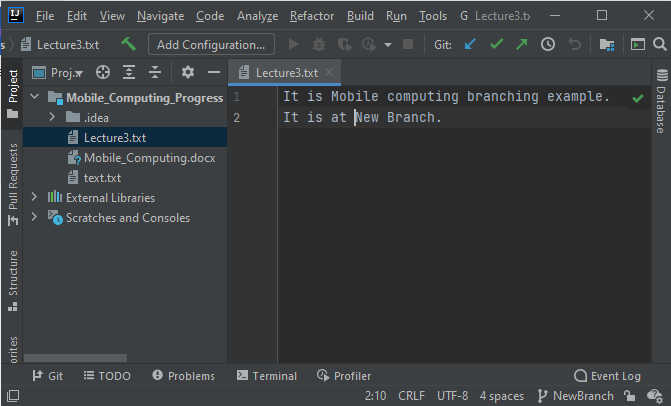
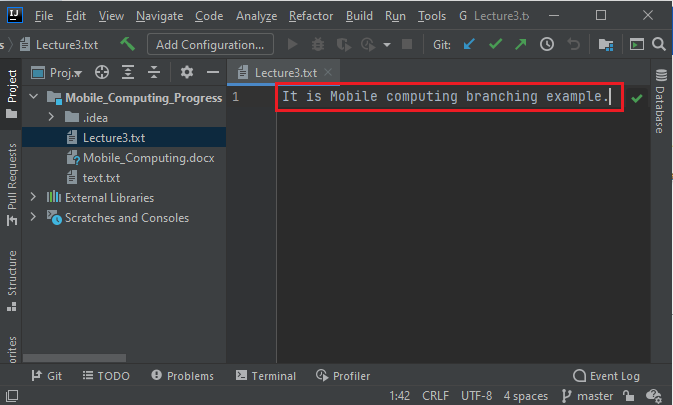
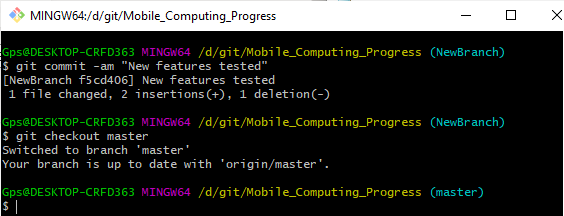
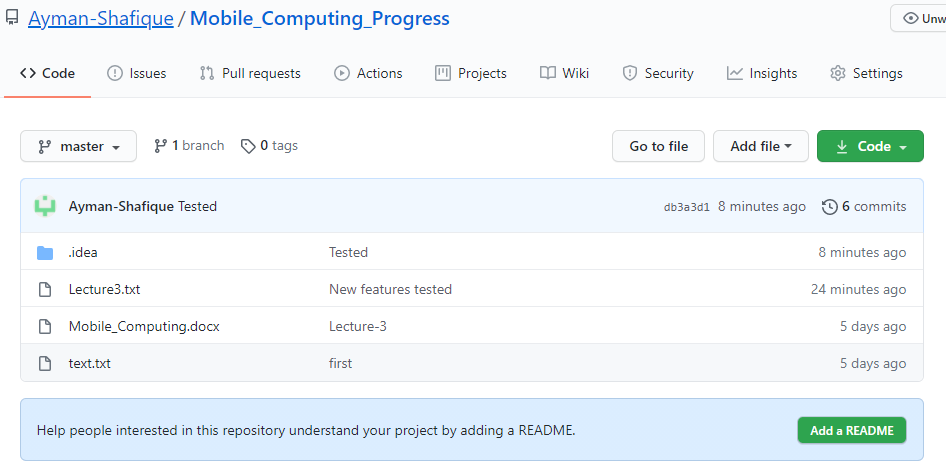
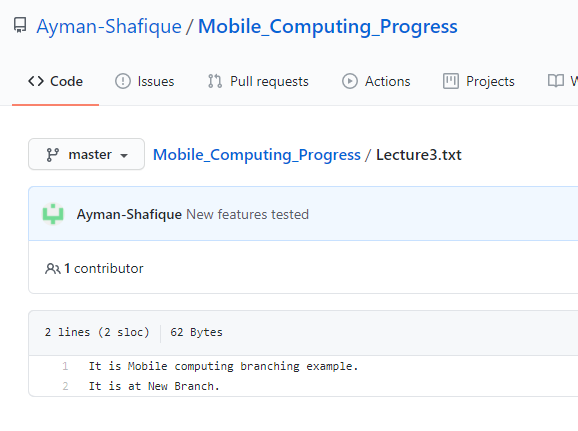


* **Merge issue resolved now text.txt is:**
* **Git log:**
  + **git log:**
* **Git add:**
  + **Two new files are added:**
* **Git rm:**
  + **Git rm file3.txt:**

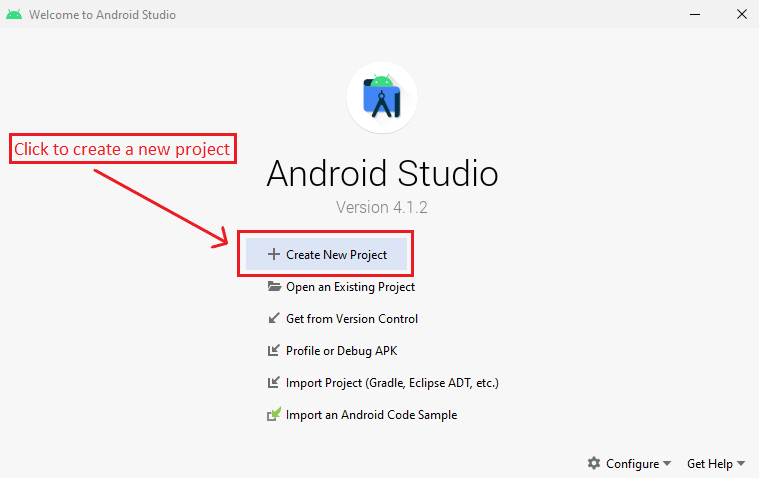
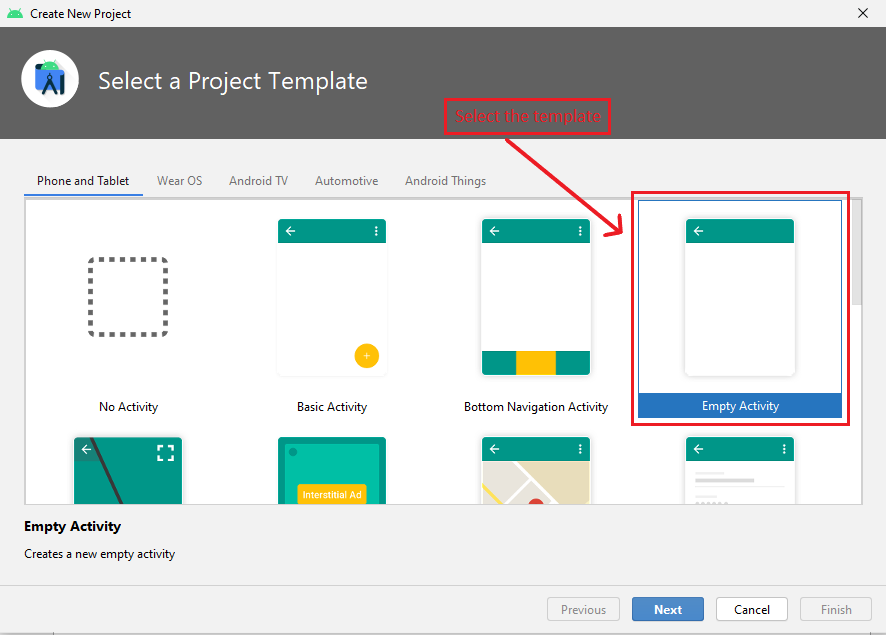


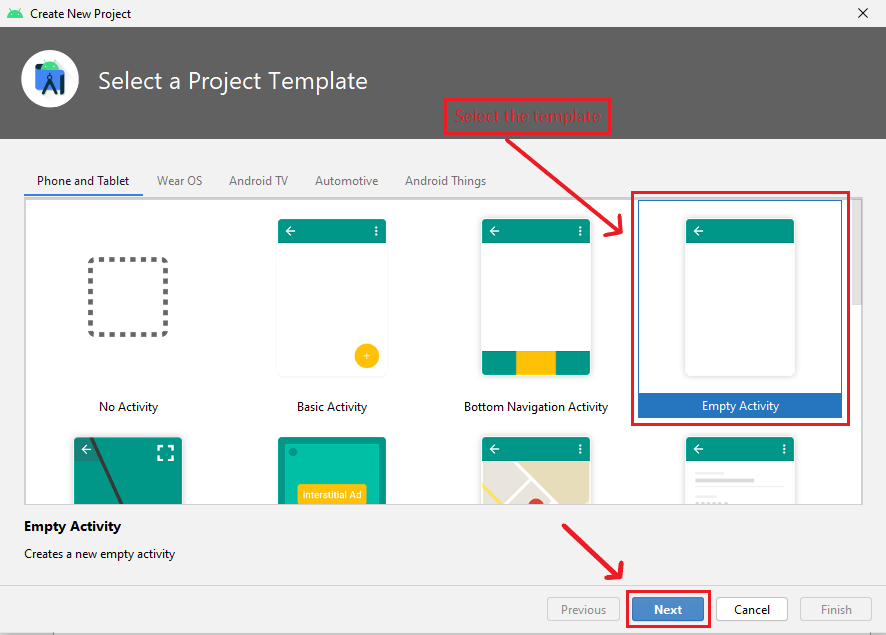
* **File3 deleted:**
* **Merge Conflict:**

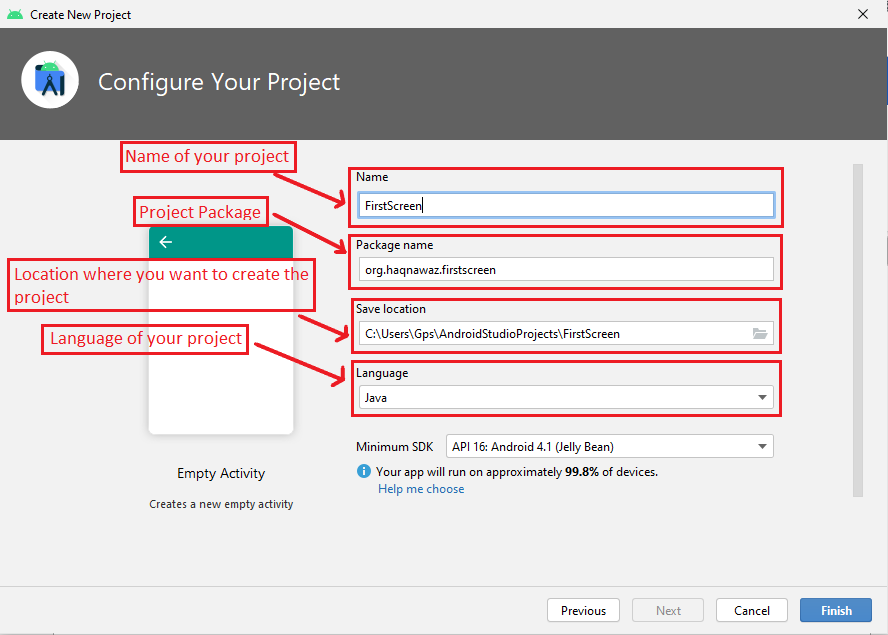
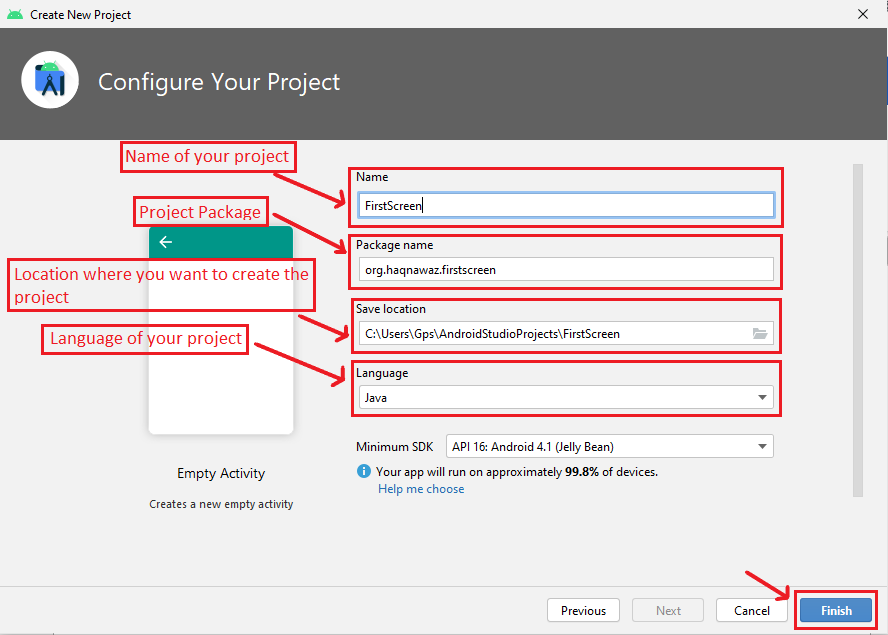
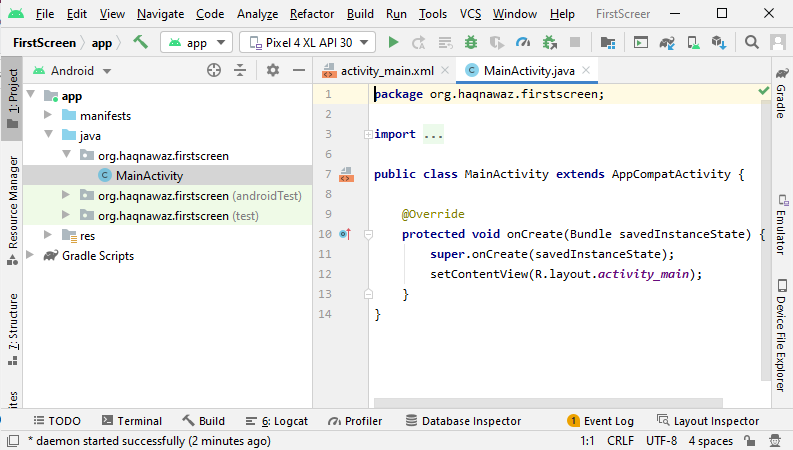
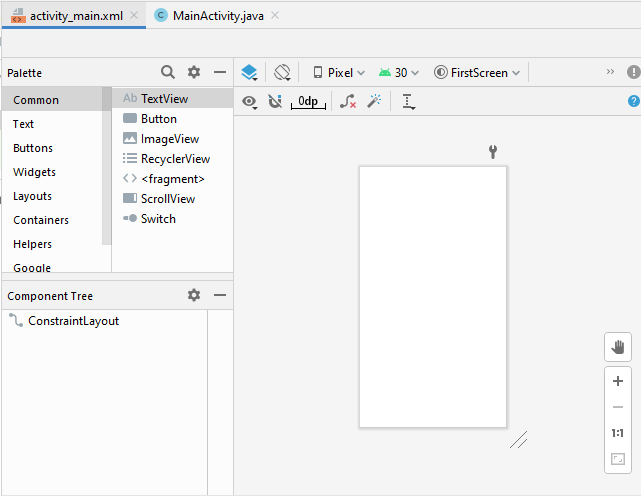


* **Git log:**
  + The **Git Log** tool allows you to view information about previous commits that have occurred in a project. The simplest version of the **log** command shows the commits that lead up to the state of the currently checked out branch. These commits are shown in reverse chronological order (the most recent commits first).
* **Branching:**
  + Lecture3.txt file:
* **Git branch:**
  + List branches (the asterisk denotes the current **branch**).
  + **Command:** git branch
* **Git checkout:**
  + The git checkout command is used to switch between branches in a repository. Be careful with your staged files and commits when switching between branches. The git checkout command operates upon three different entities which are files, commits, and branches.
  + **Command:** git branch –b “new branch name”
  + Check the branch in which we are right now:
  + Now the Lecture3:
* **Git checkout:**
* **Git commit:**
* **File:****----------------------------- Lecture # 04 ---------------------------**

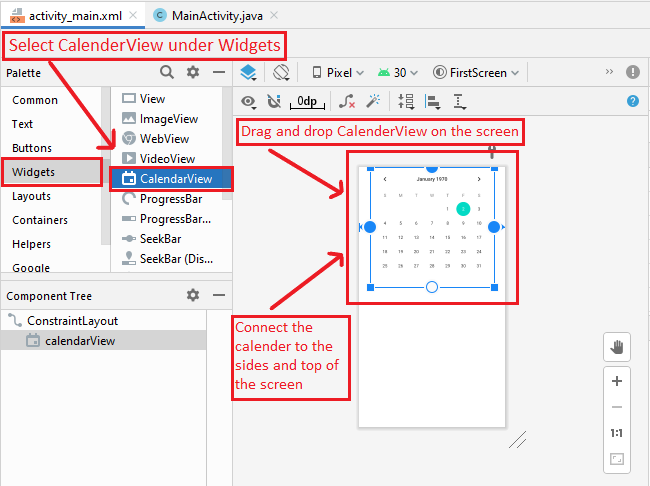
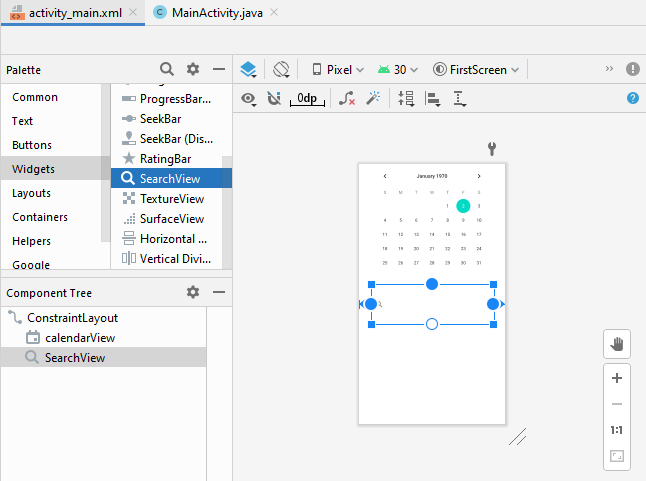
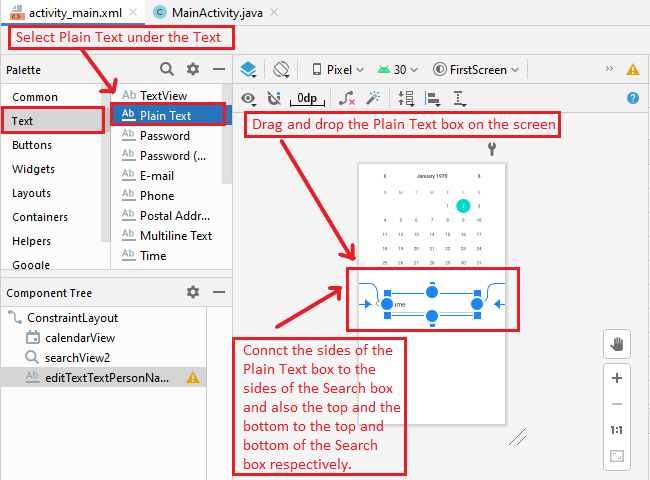
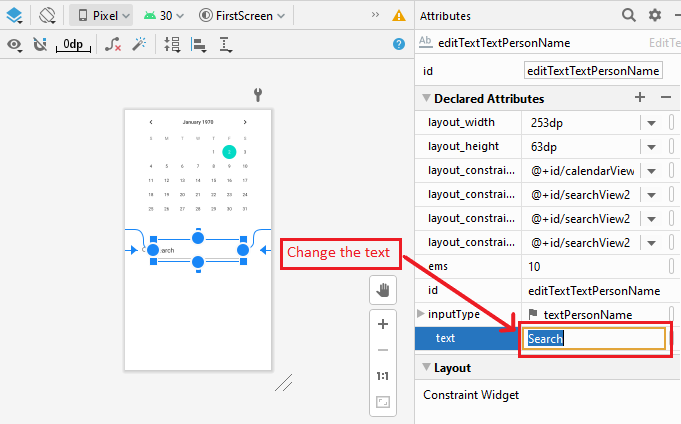
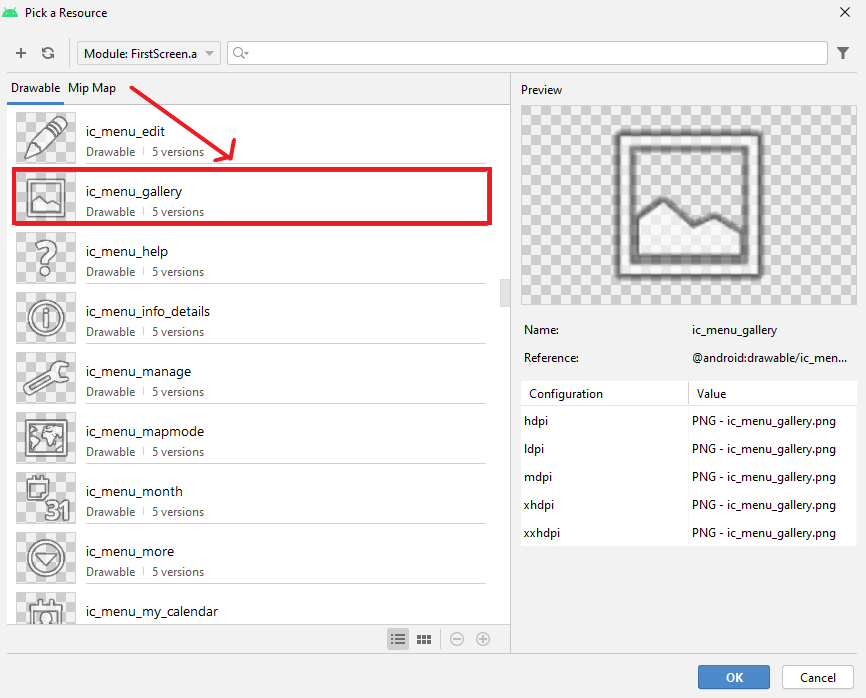
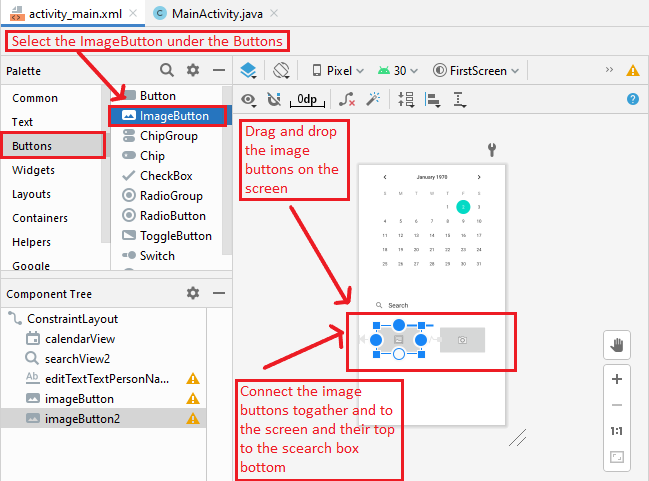
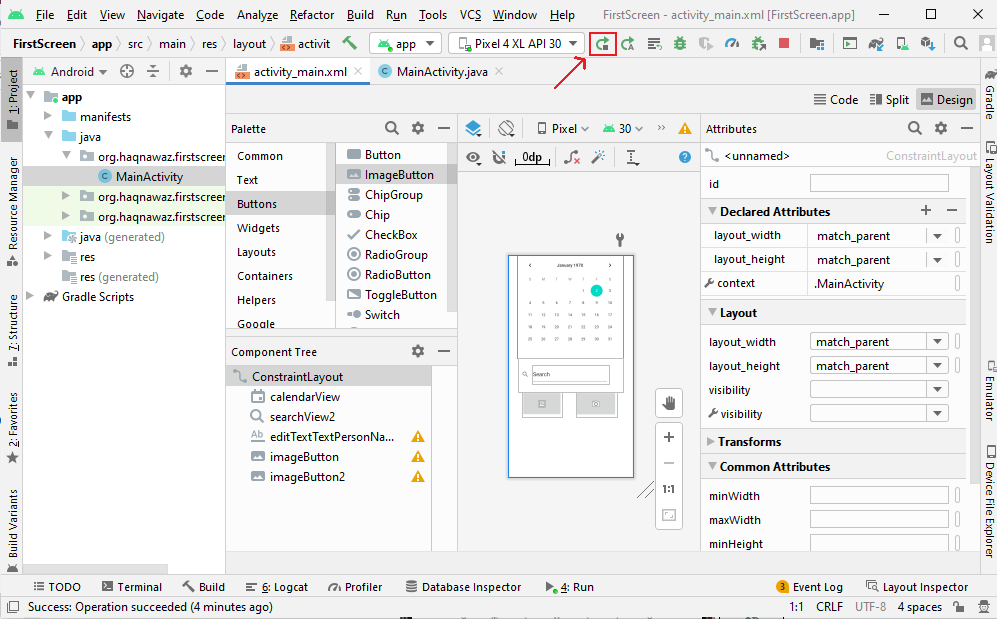
**Android Studio**

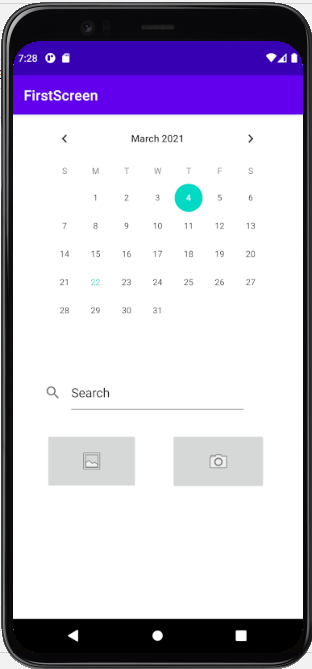
* **Start a new project:**
* **Select a template:**

**Then enter next**  

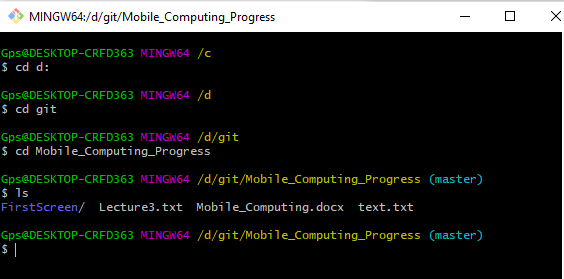
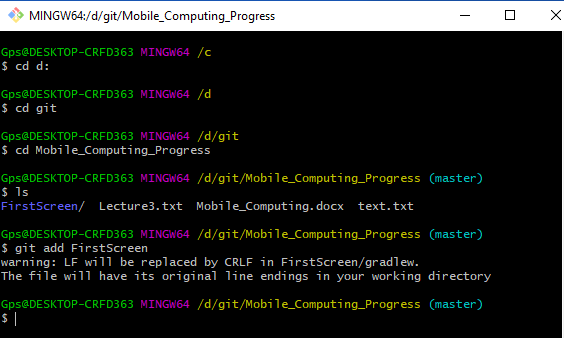
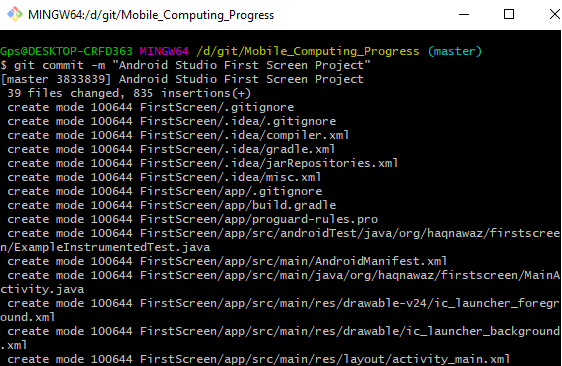
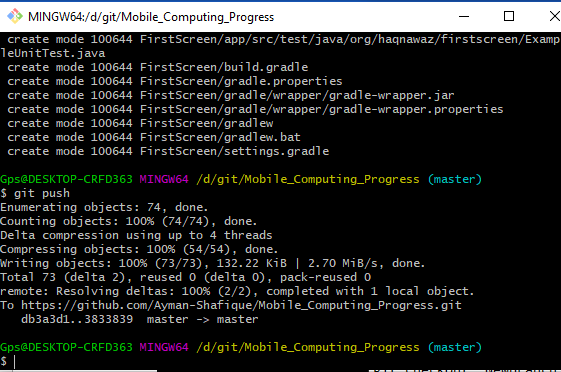
* **Project Specifications:** **Then enter next** 
* **Project View:**
* **Go to activity\_main\_xml:**

**Creating A Screen ON Android Studio**

* **Calendar View:**
  + Select the Widget.
  + Select the CalendarView.
  + Drag the CalendarView and drop on the screen.
  + Connect the top of CalendarView to the top of the screen.
  + Connect the right and left side of CalendarView to the right and left side of screen respectively.
* **SearchView:**
  + Select the Wigets.
  + Select the SearchView.
  + Drag the SearchView and drop on the screen.
  + Adjust the size.
  + Connect the top of SearchView to the bottom of the Calendar.
  + Connect the right and left side of SearchView to the right and left side of screen respectively.
* **Plain Text:**
  + Select the Text.
  + Select the Plain Text.
  + Drag the Plain Text and drop on the screen.
  + Adjust the size.
  + Connect the top and bottom of PlainText box to the top and bottom of the SearchView box respectively.
  + Connect the right and left side of Plain Text to the right and left side of SearchView box respectively.
* **Change the Text in Plain Text Box:**
  + Click on the Plain Text Box.
  + Go to attributes.
  + Change the text in **text** column.
* **Image Buttons:**
  + Select the Buttons.
  + Select the ImageButton.
  + Drag the ImageButton and drop on the screen.
  + Select the ImageButton you want to use.
  + Adjust the size.
  + Connect the top ImageButton to the top and bottom of the SearchView box.
  + Connect the two ImageButtons togather.
  + Connect the right and left side of Plain Text to the right and left side of SearchView box respectively.
* **Run the Project:**
* **The Screen:**



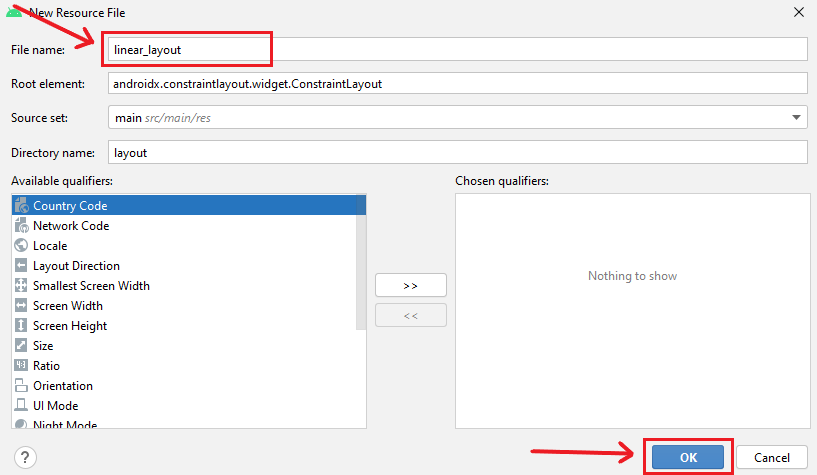
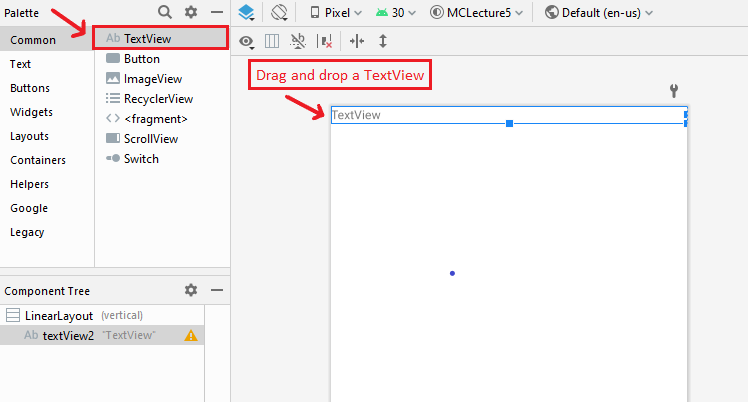
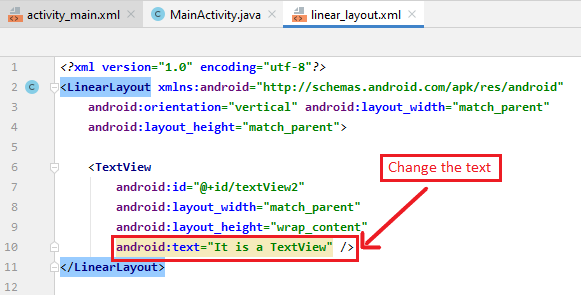
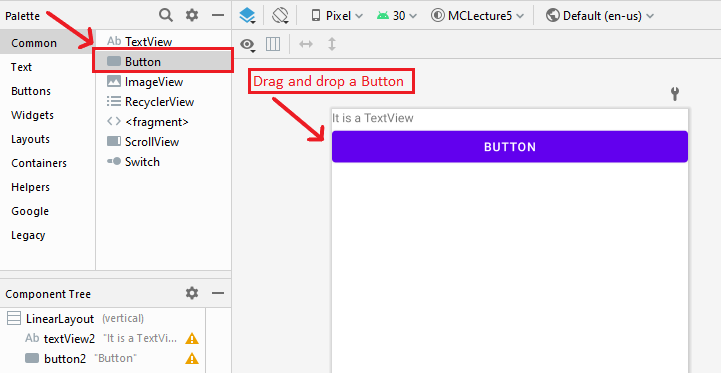
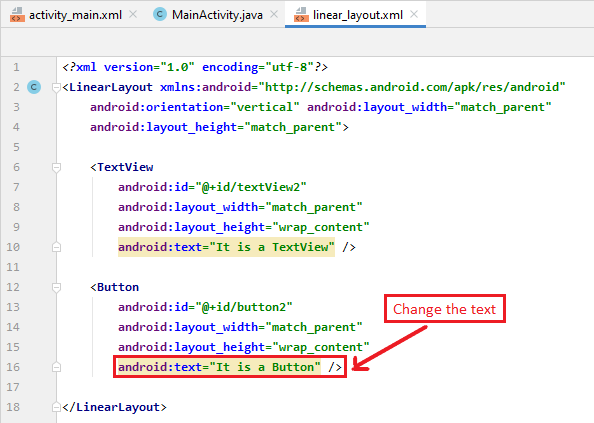
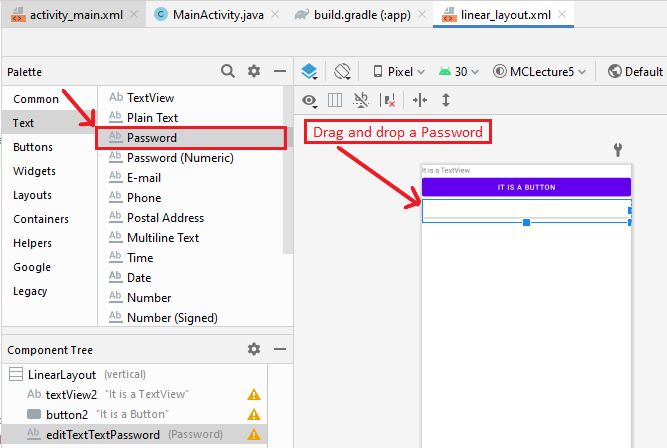
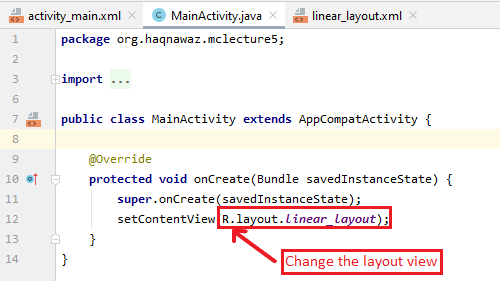
**Uploading the Project on GitHub**

* **Check the folder:** 
* **Git add:**
* **Git commit:**
* **Git push:**

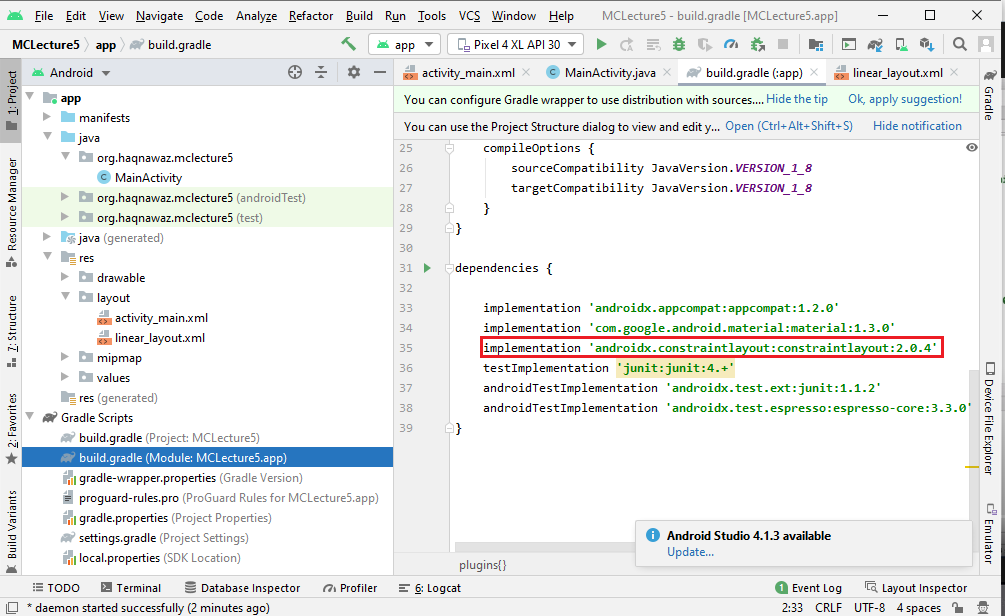
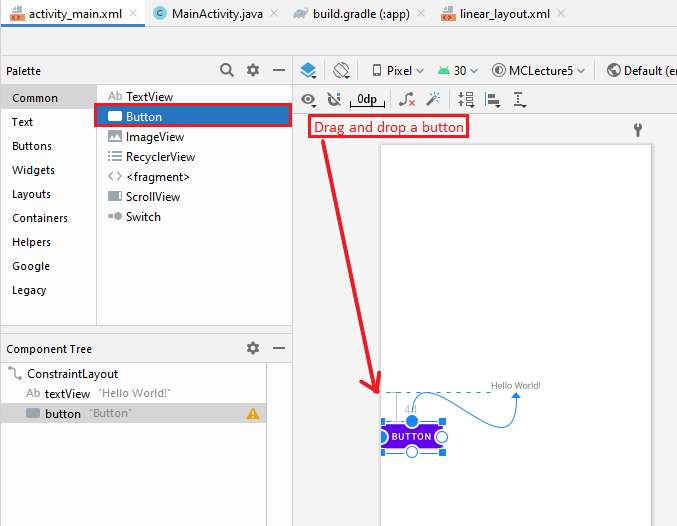
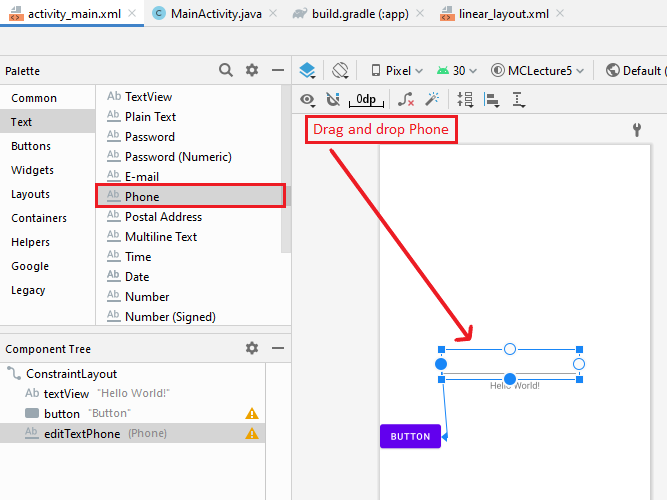
**----------------------------- Lecture # 05 ---------------------------**

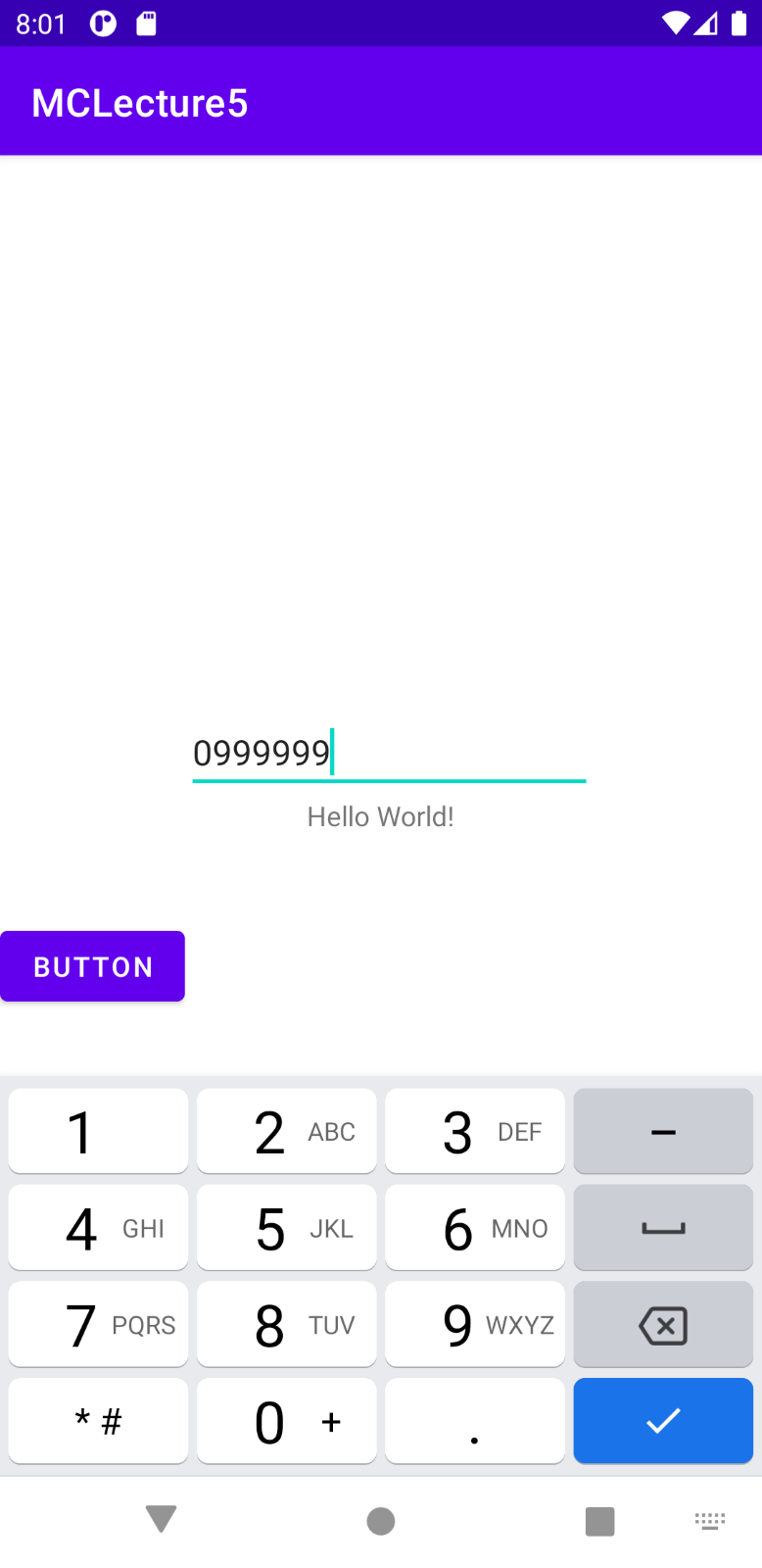
* **View:**
  + A view usually draws something the user can see and interact with. Physical thing that user can touch such as buttons, touch boxes.
  + The View objects are usually called **Widgets** and can be one of many subclasses, such as **Button** or **TextView**.
* **ViewGroup:**
  + Invisible container
  + The ViewGroup objects are usually called **layouts** can be one of many types that a different layout structure, such as **LinearLayout** or **ConstraintLayout**.

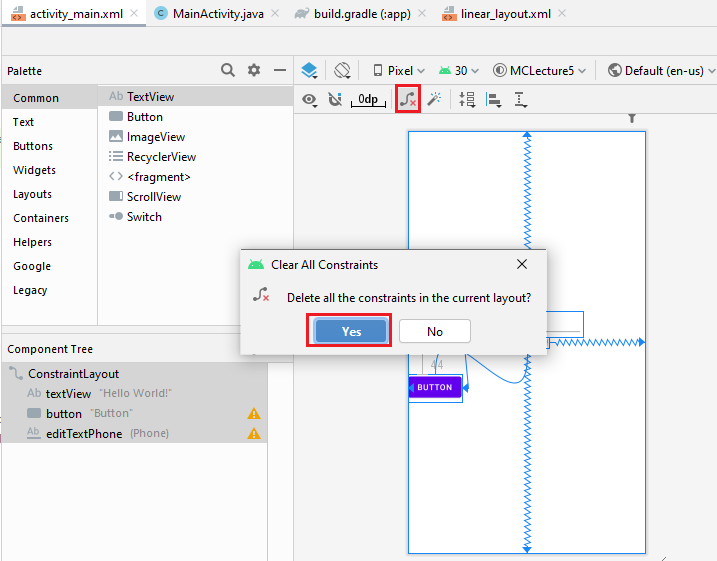
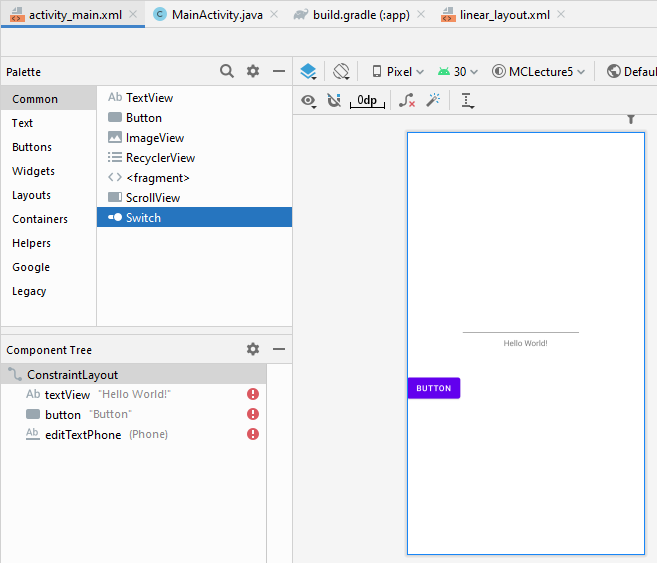
**Linear Layout**

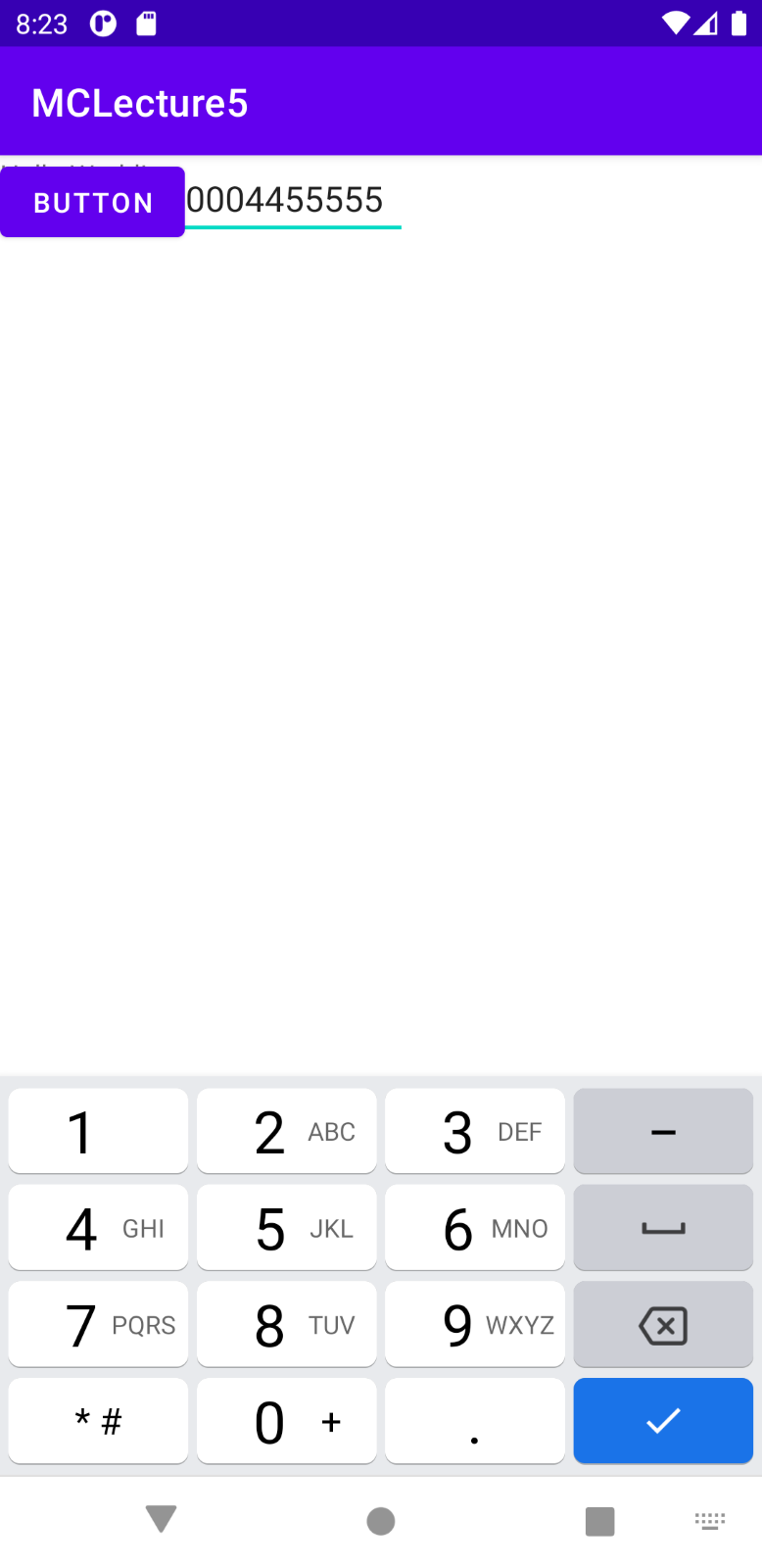
* **Adding a new layout:**
  + **Adding a TextView:**
    - Change the text.
  + **Adding a Button:**
    - Change the text.
  + **Adding a Password:**
  + **linear\_layout.xml:**
* **MainActivity.java** 
  + Change the layout inMainActivity.java
  + **Run App:**

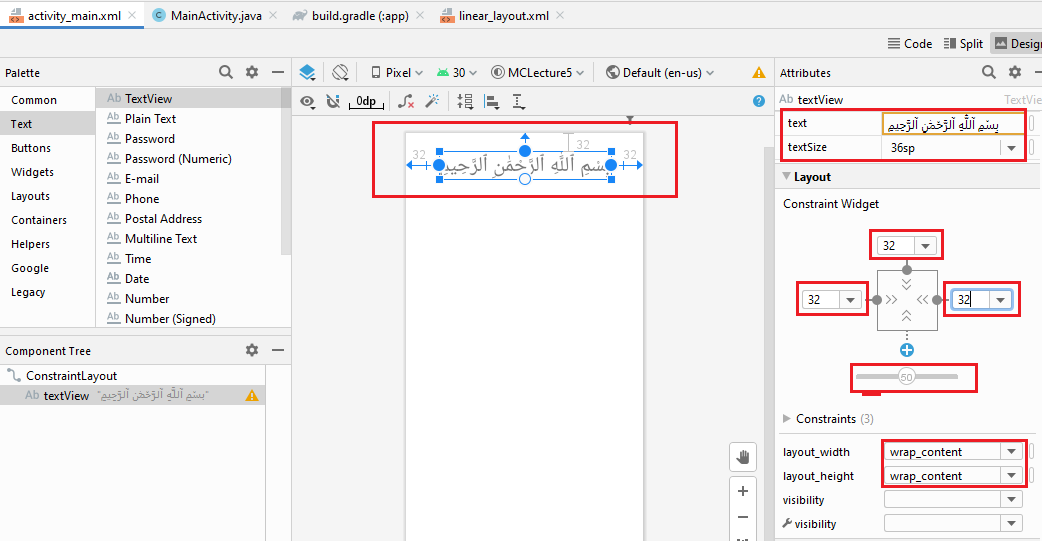


* **Dependency:**
* **Activity\_main.xml:**
  + **Adding a Button:**
    - Drag and drop the Button.
    - Set its tops constraint to the bottom of the text.
    - Set its left constraint to the side of the screen.
  + **Adding a Phone:**
    - Drag and drop the Phone under the Text.
    - Set its bottom constraint to the top of the text.
    - Set its left constraint to the right of the button.
  + **Run App:**

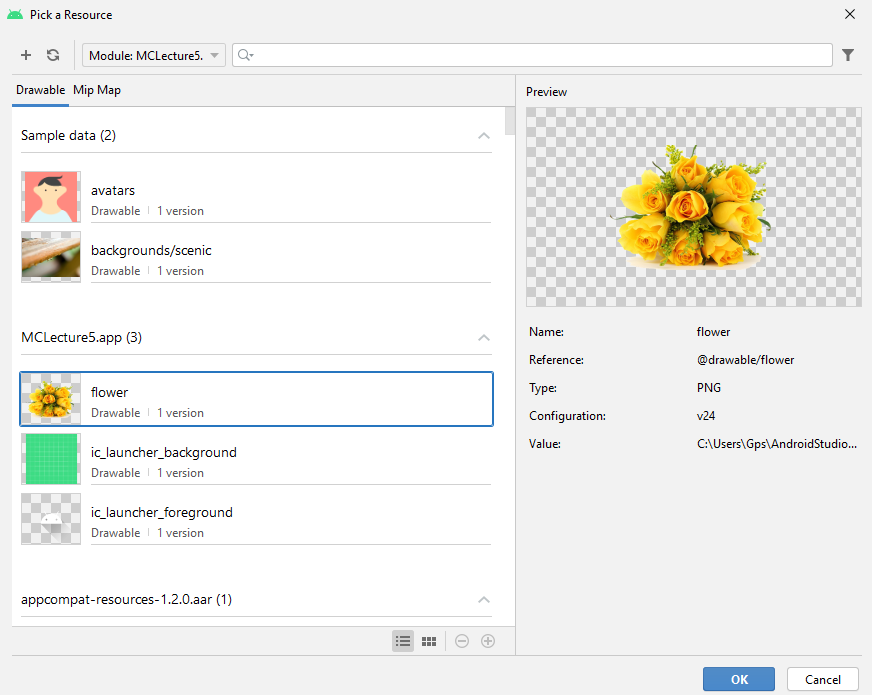
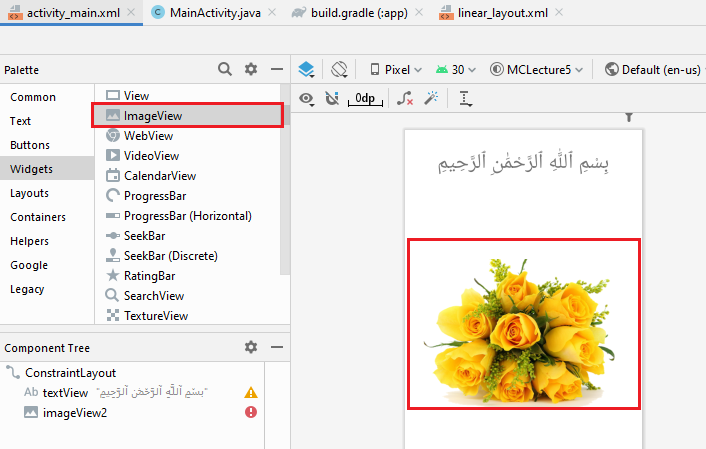


* + **Remove Constraints:**
    - Click the clear all constraints button on the top.
    - **After clearing all constrainst:**
  + **Run App:**

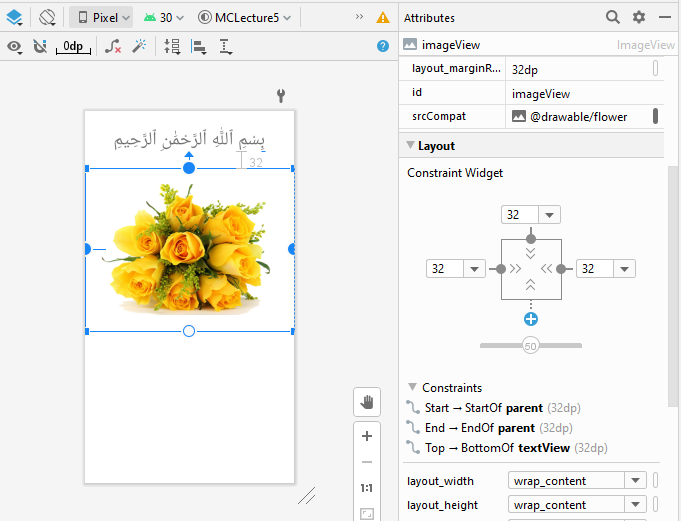


* + **TextView:**
    - Add a TextView by drag and drop.
    - Add some text.
    - Set its constraints.
    - Set its baise.
  + **Run App:**

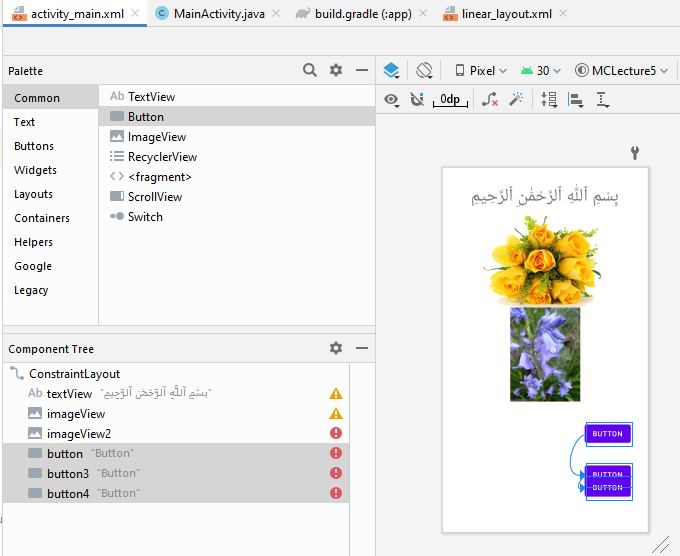
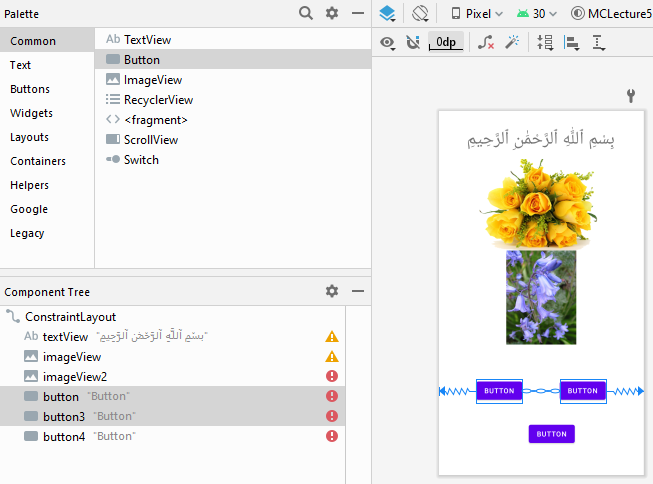
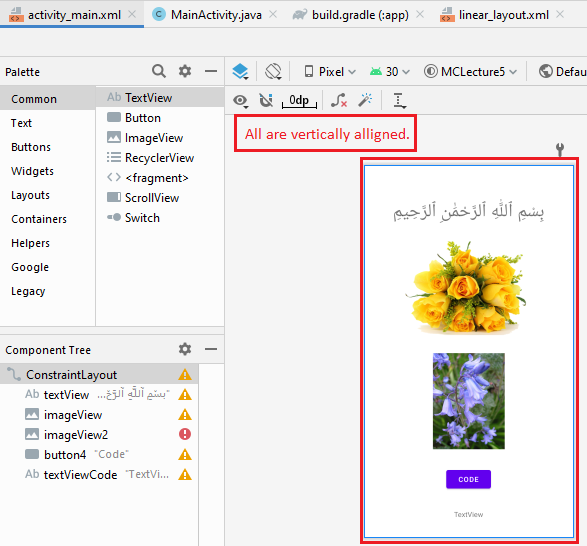
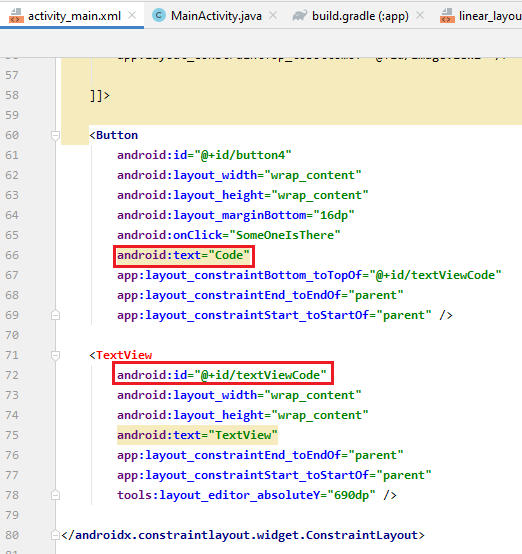
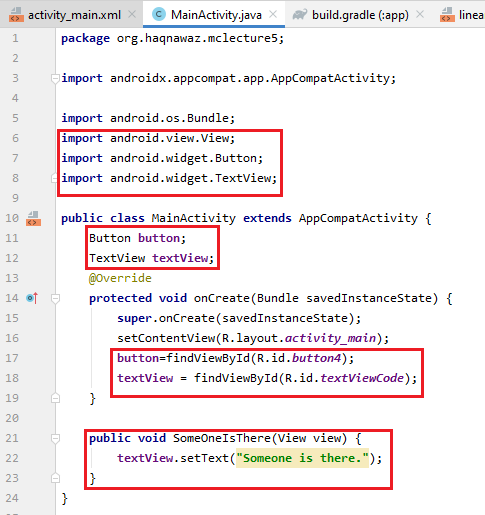


* + **Adding an ImageView:**
    - Drag and drop an ImageView.
    - Select the image.
    - Click ok:
  + **Run App:**



* + **Setting Constraints:**
    - Set the image top constraint to the bottom of the textView.
  + **Run App:**



* + 
  + 
* **On click Button:**
  + **Activity\_main.xml:**
  + **Code:**  
  + **MainActivity.java:**
  + **Run App:**



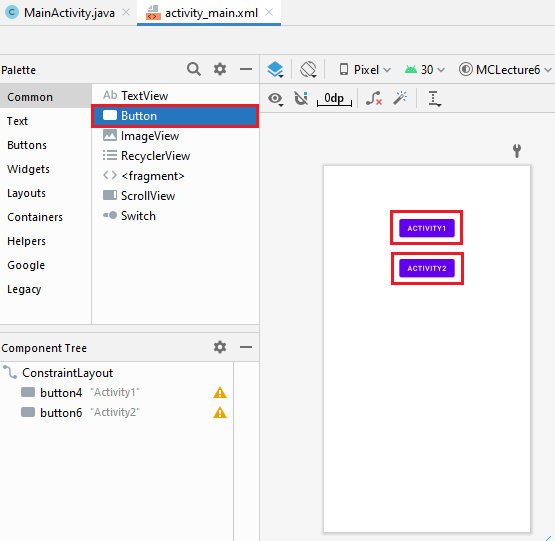
* + - On clicking the button



**----------------------------- Lecture # 06 ---------------------------**

**Working With Multiple Activities**

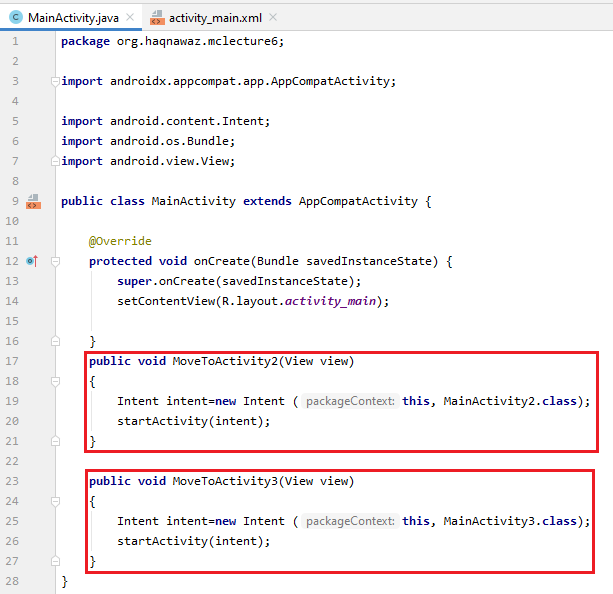
* + **Activity\_main.xml:**
    - **Adding Buttons:**
      * Add two buttons
      * Drag and drop buttons
      * Set their constraints
      * Set their text as Activity1 and Activity2



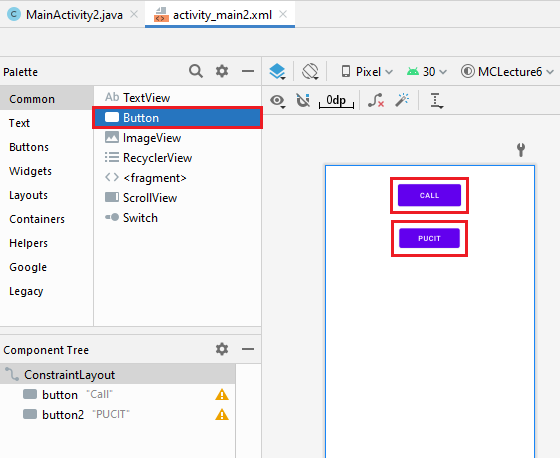
* + - **Button Activity1:**
      * On click button Activity1 the new **MainActivity2.java** is called.
    - **Button Activity2:**
      * On click button Activity2 the new **MainActivity3.java** is called.



* + **MainActivity.java:**
    - Method **MoveToActivity2()** calls the new Activity the MainActivity2.java.
    - Method **MoveToActivity3()** calls the new Activity the MainActivity3.java.



* + **Activity\_main2.xml:**
    - **Adding Buttons:**
      * Add two buttons
      * Drag and drop buttons
      * Set their constraints
      * Set their text as Activity1 and Activity2



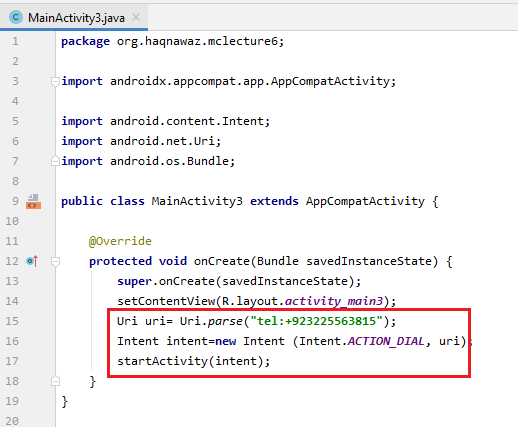
* + - **Button Call:**
      * On click button Call the function **Call()** is called.
    - **Button PUCIT:**
      * On click button PUCIT the function **showWebsite()** is called.



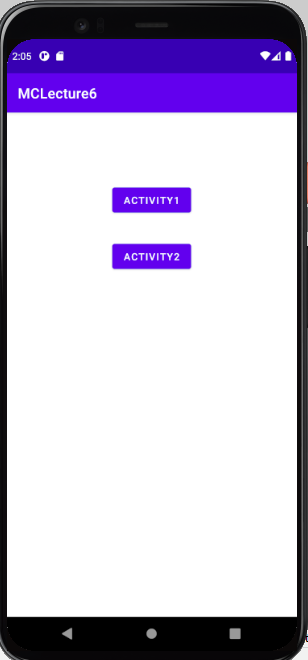
* + **MainActivity2.java:**
    - Method **Call()** calls the new Activity for call.
    - Method **showWebsite()** calls the new Activity for the website of pucit.



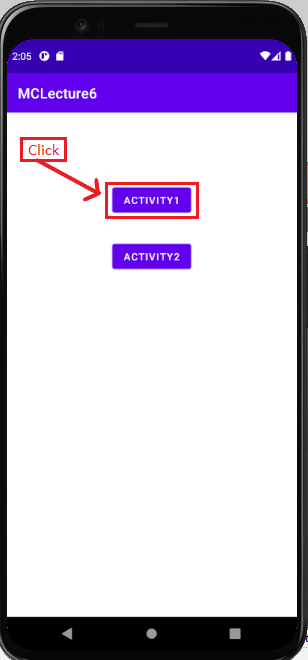
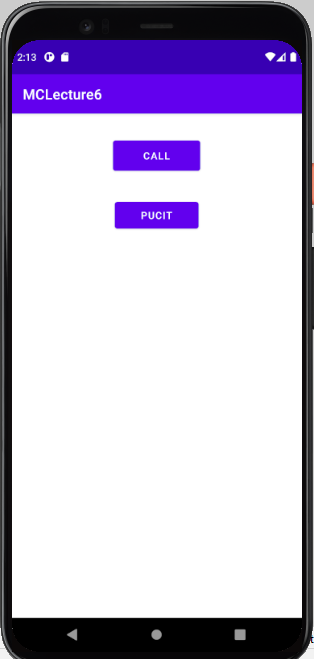
* + **MainActivity3.java:**
    - On running the activity, the call activity is called.



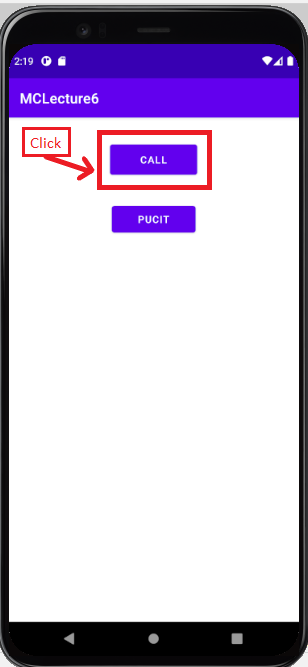
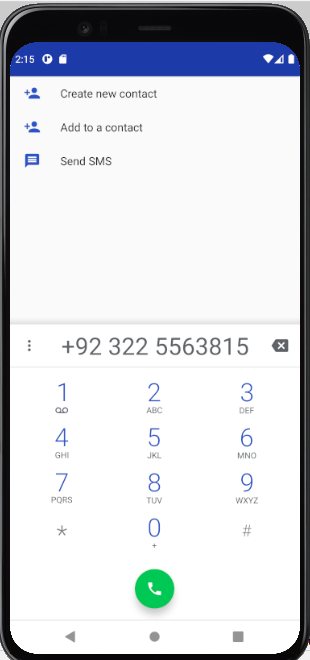
* + **Run App:**



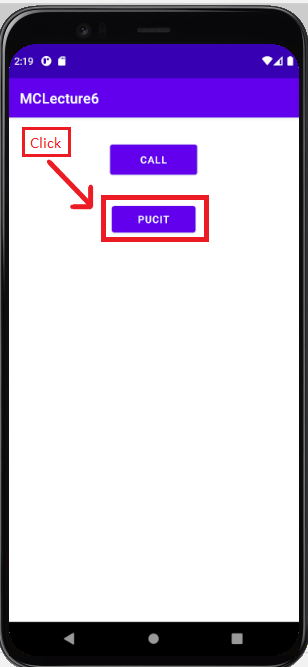
* + - **Clicking Activity1:**

* + - **Clicking Call:**

* + - **Clicking PUCIT:**

* + - **Clicking Activity2:**