**( الرَّحِيمِ الرَّحْمَٰنِ اللَّهِ بِسْمِ )**

**MOBILE COMPTUING**

**Teacher : Sir Haq Nawaz**

**Name : Umer Khan**

**Std Id : BSEF18A020**

**Mobile computing:**

****

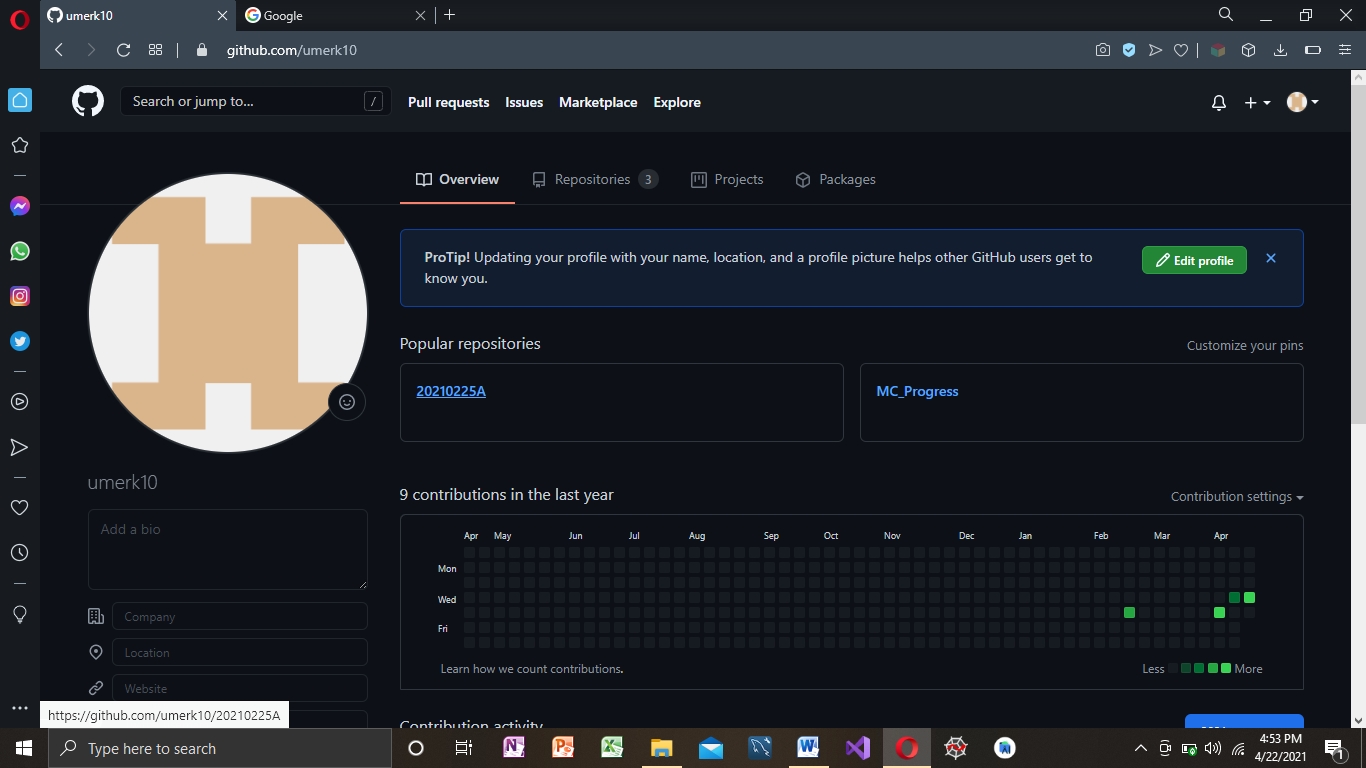
**Mobile computing** is [human–computer interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction) in which a [computer](https://en.wikipedia.org/wiki/Computer) is expected to be transported during normal usage, which allows for the transmission of data, voice, and video. Mobile computing involves mobile communication, mobile hardware, and mobile software. Communication issues include [ad hoc networks](https://en.wikipedia.org/wiki/Mobile_ad_hoc_network) and infrastructure networks as well as communication properties, [protocols](https://en.wikipedia.org/wiki/Communications_protocol), data formats, and concrete technologies. Hardware includes [mobile devices](https://en.wikipedia.org/wiki/Mobile_device) or device components. Mobile software deals with the characteristics and requirements of mobile applications.

**Git Hub:**

****

**GitHub, Inc.** is a provider of [Internet hosting](https://en.wikipedia.org/wiki/Internet_hosting_service) for [software development](https://en.wikipedia.org/wiki/Software_development) and [version control](https://en.wikipedia.org/wiki/Version_control) using [Git](https://en.wikipedia.org/wiki/Git" \o "Git). It offers the [distributed version control](https://en.wikipedia.org/wiki/Distributed_version_control) and [source code management](https://en.wikipedia.org/wiki/Source_code_management) (SCM) functionality of Git, plus its own features. It provides [access control](https://en.wikipedia.org/wiki/Access_control) and several collaboration features such as [bug tracking](https://en.wikipedia.org/wiki/Bug_tracking_system), [feature](https://en.wikipedia.org/wiki/Software_feature) requests, [task management](https://en.wikipedia.org/wiki/Task_management), [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration) and [wikis](https://en.wikipedia.org/wiki/Wiki) for every project

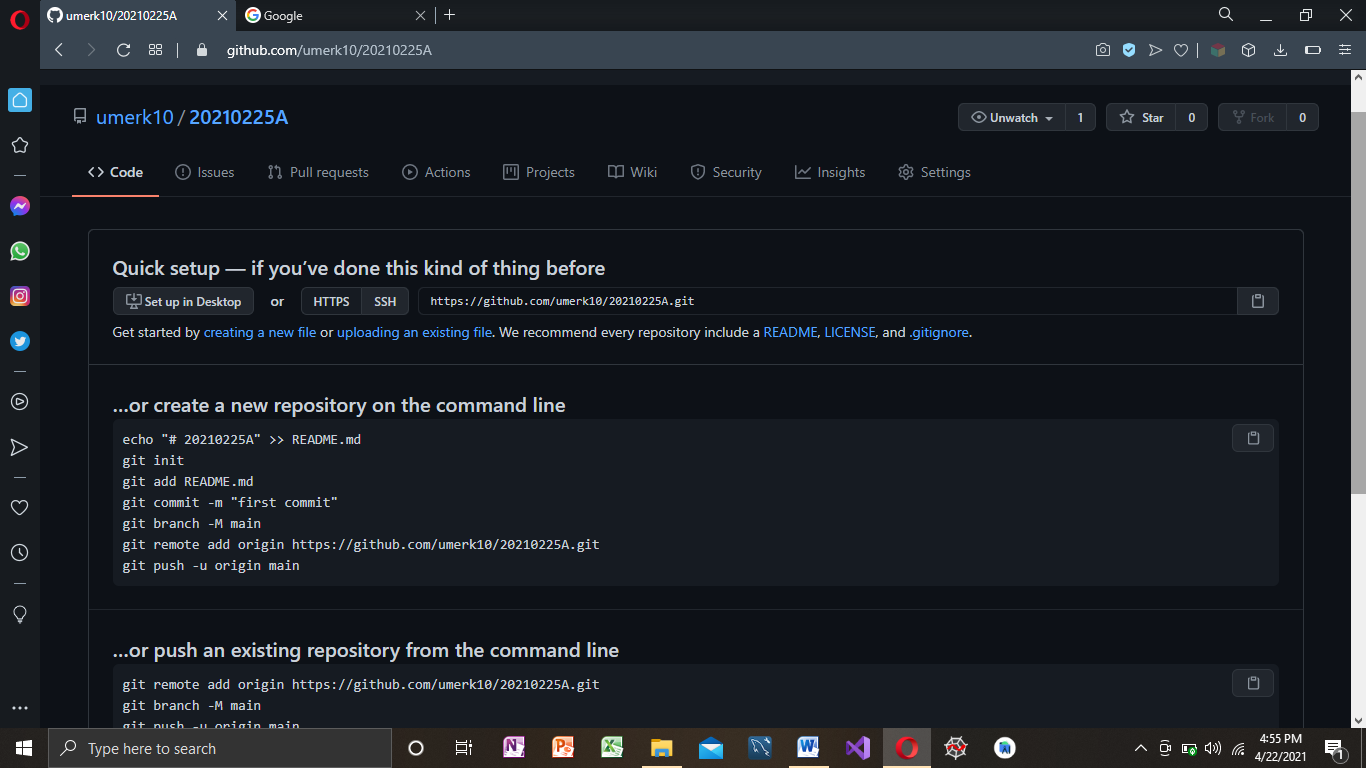
Git Hub account **: umerk10**



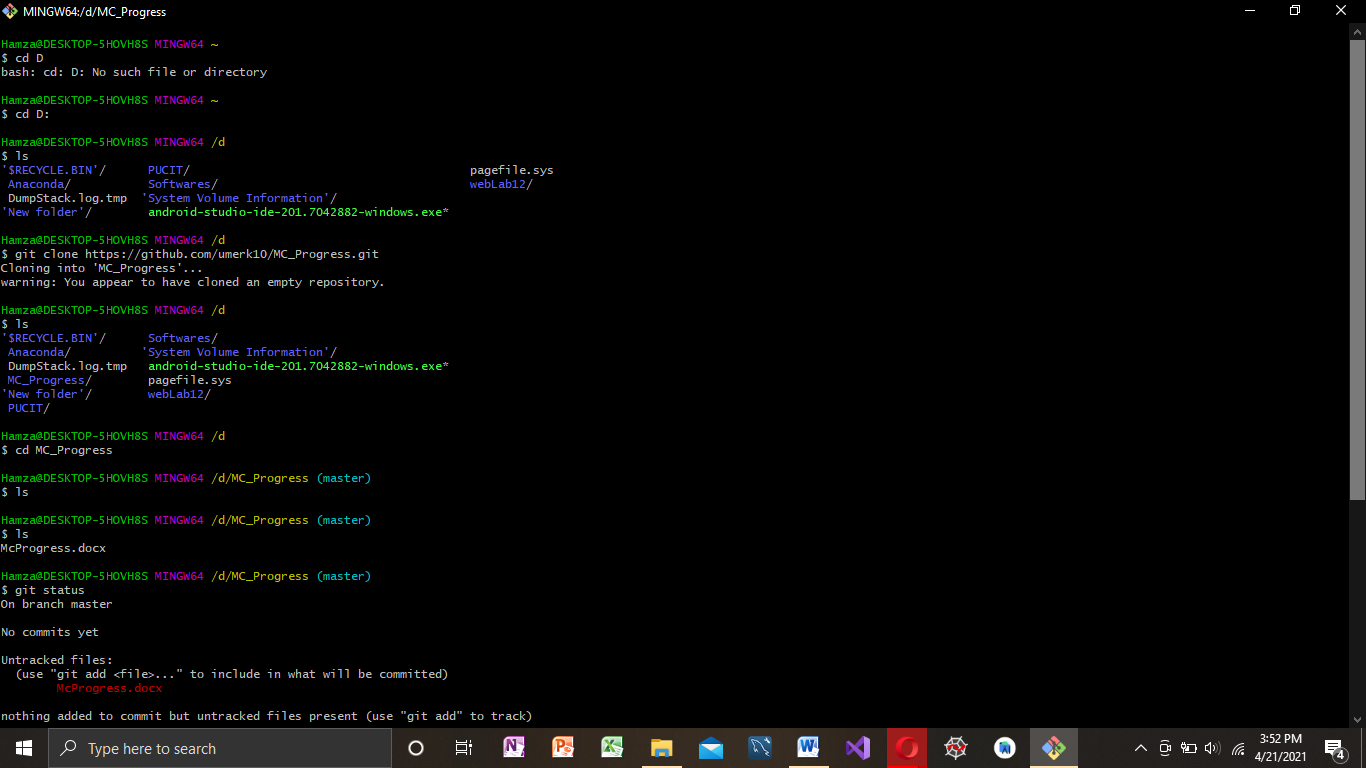
1. **Step** 0: **Install** git and create a **GitHub** account. ...
2. **Step** 1: Create a local git repository. ...
3. **Step** 2: Add a new file to the repo. ...
4. **Step** 3: Add a file to the staging environment. ...
5. **Step** 4: Create a commit. ...
6. **Step** 5: Create a new branch. ...
7. **Step** 6: Create a new repository on **GitHub**. ...
8. **Step** 7: Push a branch to **GitHub**.

Repsitory Making :

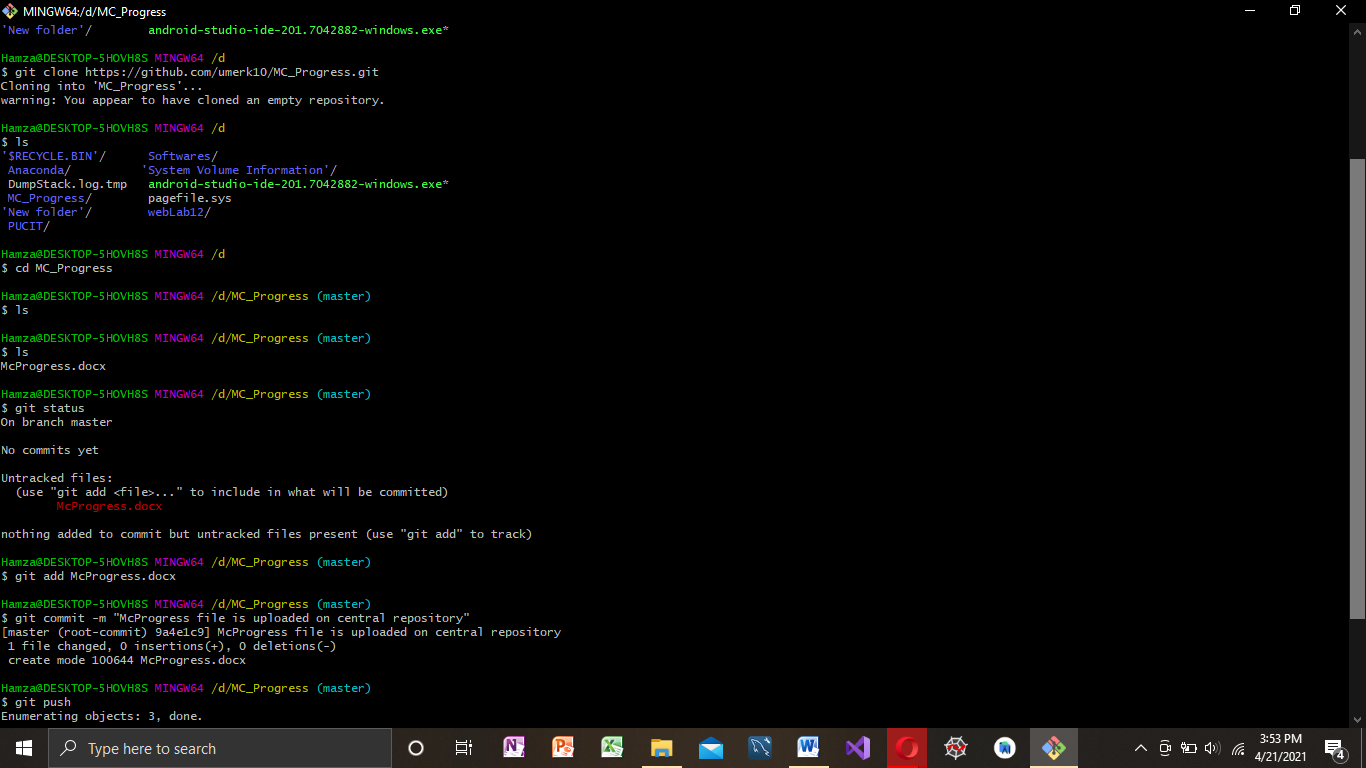
https://github.com/umerk10/20210225A.git

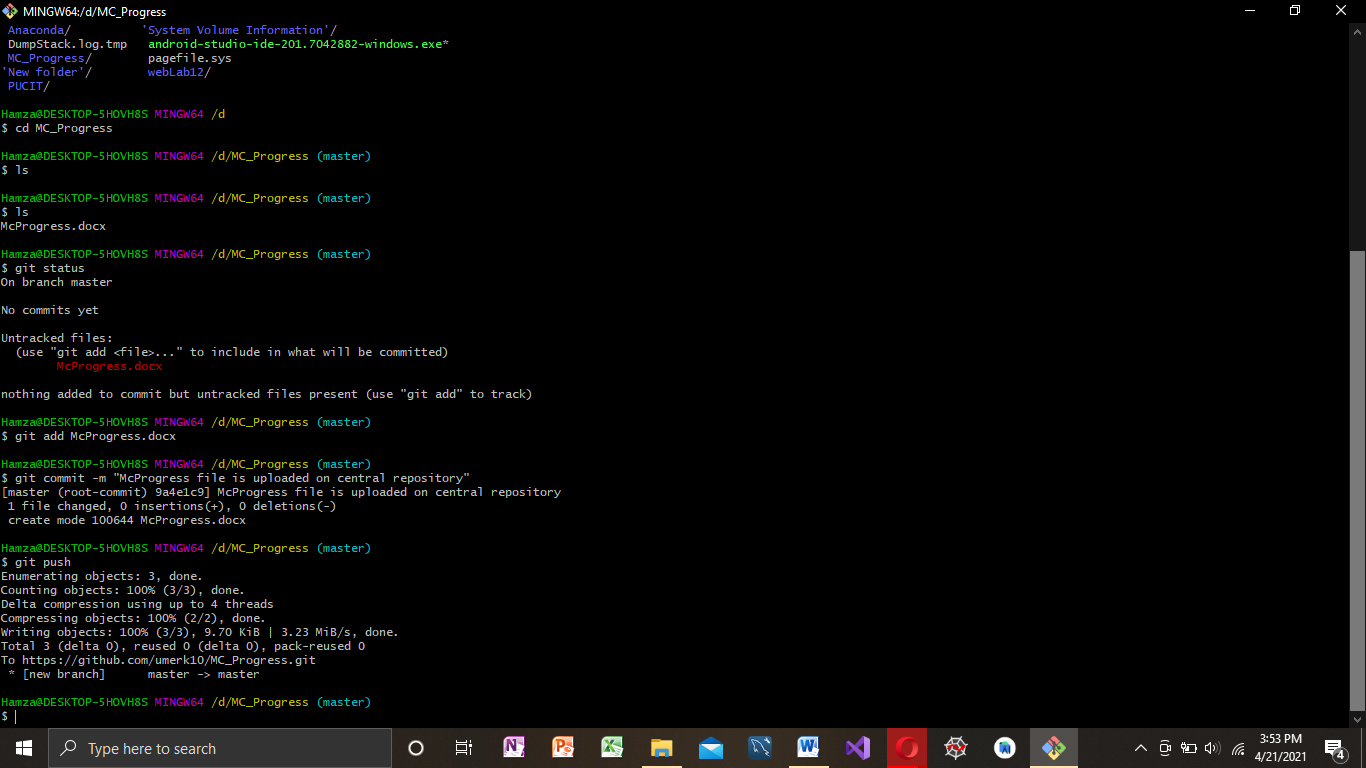


Git clone



Git Add,Commit,Push,Status:





**Andriod Studio:**



**Android Studio** is the official[[7]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-:0-7) [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains" \o "JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development).[[8]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-8) It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems or as a subscription-based service in 2020.[[9]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-9)[[10]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-10) It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (E-ADT) as the primary IDE for native Android application development.

Installation Steps:

##### Step 1: Install "Android Studio IDE"

**Reference**: "Install Android Studio" @ <https://developer.android.com/studio/install>.

##### Step 2: Installing Android SDK

This step takes a long time as you need to download about 3GB of zip data, and expand to 5 GB of disk data, even for the minimum configuration.

Project Creating Steps:

##### Step 0: Read

Goto "Android Guides" @ <https://developer.android.com/guide/index.html>. Read "Build your first app".

##### Step 1: Create a New Android Project

1. Launch Android Studio.
2. Choose "Start a new Android Studio Project".
3. In "Select a Project Template" ⇒ select "Phone and Tablet" tab ⇒ "Empty Activity" ⇒ Next.
4. In "Configure your project" ⇒ Set "Name" to "Hello Android" (this will be the "Title" in your phone's app menu) ⇒ The "Package name" and "Save Location" will be updated automatically ⇒ In "Language", select "Java" ⇒ Leave the "Minimum API Level" and the rest to default ⇒ Finish.
5. Be patient! It could take a few minutes to set up your first app. Watch the "progress bar" at the bottom status bar and Zzzzzzzzz...... If you see "2 processes running", you can click on it to see the details. Once the progress bar indicates completion, **a hello-world app is created by default**.

##### Step 2: Setup Emulator (aka Android Virtual Device or AVD)

To run your Android app under an emulator, you need to first create an Android Virtual Devices (AVD). An AVD emulates a specific device (e.g., your zPone 99 or Taimi Green). You can create AVDs to emulate different devices (e.g., phone/tablet, android version, screen size, and etc.).

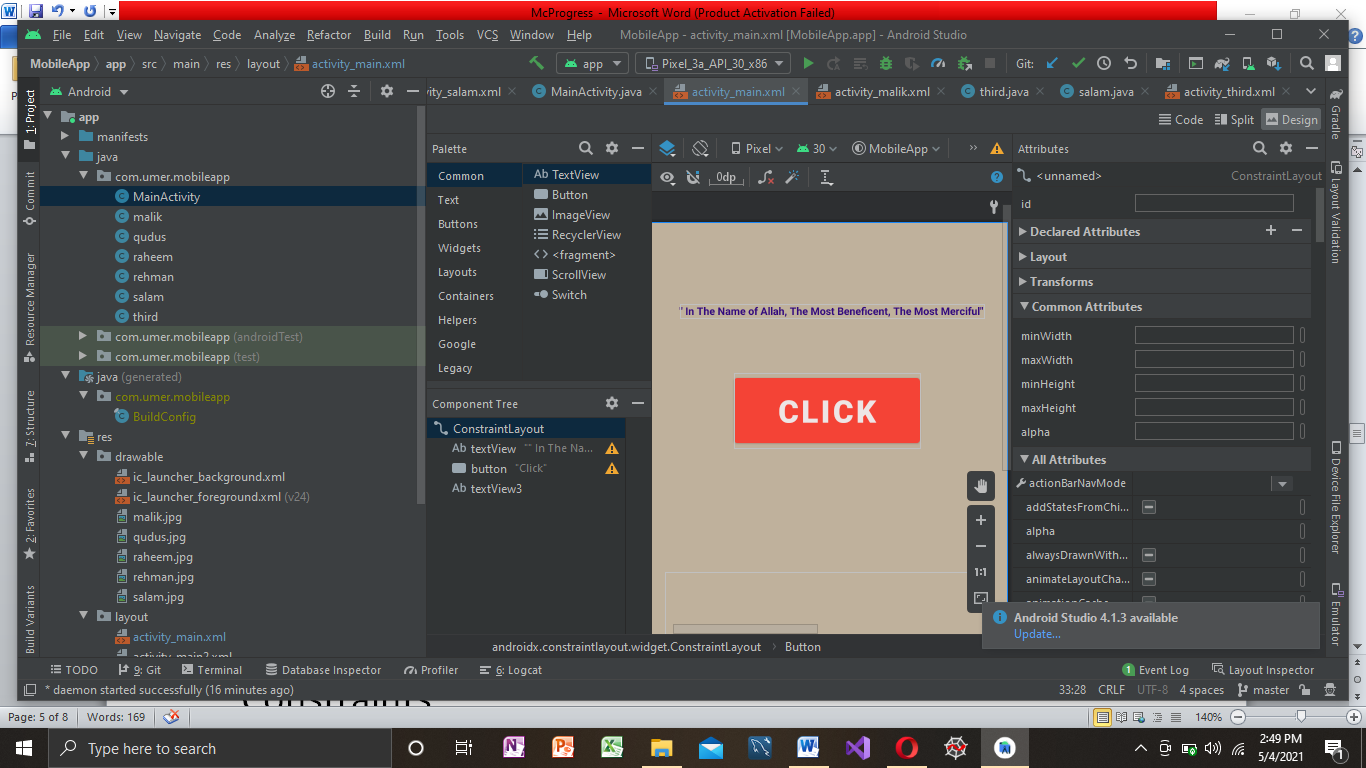
1. In Android studio, select "Tools" ⇒ Android ⇒ AVD Manager. See "Common Errors" below if you cannot find "AVD manager".
2. (New) A Virtual Device called "Pixel\_3a\_API\_30\_x86" has been created during the installation. Skip the rest of the steps (unless you want to create your own virtual devices).
3. Click "Create Virtual Device".
4. In "Choose a device definition" ⇒ In "Category", choose "Phone" ⇒ In "Name", choose "Pixel XL" or anyone that there is NO icon under "Play Store" ⇒ Next.
5. In "Select a System Image" ⇒ Use the default (without "Download" unless you want to wait for another hour) ⇒ Next.
6. In "Verify Configuration" ⇒ Finish.
7. If you see "VT-x is disabled in BIOS": Check your BIOS setting to ensure that "Virtualization Technology" is enabled. Shutdown and re-boot your PC to enter the BIOS setup. This is machine dependent. Google "*Your-PC-brand-and-model* enter BIOS setup". For example, for my HP computer ⇒ Boot ⇒ "ESC" to enter BIOS setup ⇒ Advanced ⇒ System Options ⇒ Check "Virtualization Technology (VTx)" ⇒ Save ⇒ Exit.

##### Step 3: Run the Android App on Emulator

1. Select the "Run" menu ⇒ "Run app" ⇒ Under "Available Virtual Devices", select the AVD created earlier ⇒ OK.
2. You MAY BE prompted to install Intel HAXM (Hardware Accelerated Execution Manager). Follow the instruction to install HAXM.
3. Be patient! It may take a few MINUTES to fire up the app on the emulator. You first see a blank screen ⇒ wait ⇒ Google logo ⇒ wait ⇒ G ⇒ wait ⇒ "Android" ⇒ Home screen ⇒ "Hello, world!" message.  
   **If you have problem running on the emulator, I suggest you try to run on an actual Android Phone**, if you have one. Goto next step.
4. Push the "HOME" button of the emulator ⇒ In the app menu, search for "Hello Android" app that we have just installed.
5. **DO NOT CLOSE THE EMULATOR**, as it really takes a long time to start. You could always re-run the app (or run a new app) on the same emulator. Try re-run the "Hello Android" app by selecting "Run" menu ⇒ "Run app

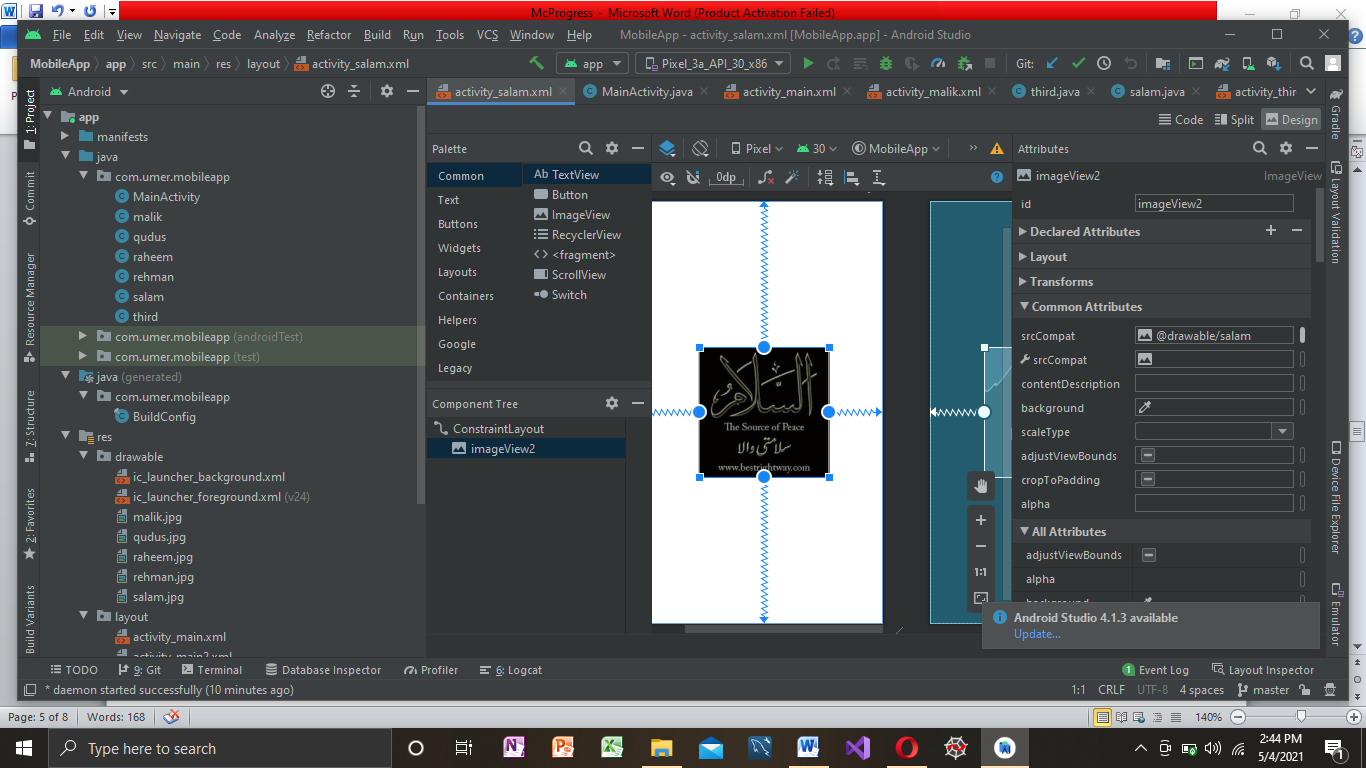
Button:

A button consists of text or an icon (or both text and an icon) that communicates what action occurs when the user touches it.

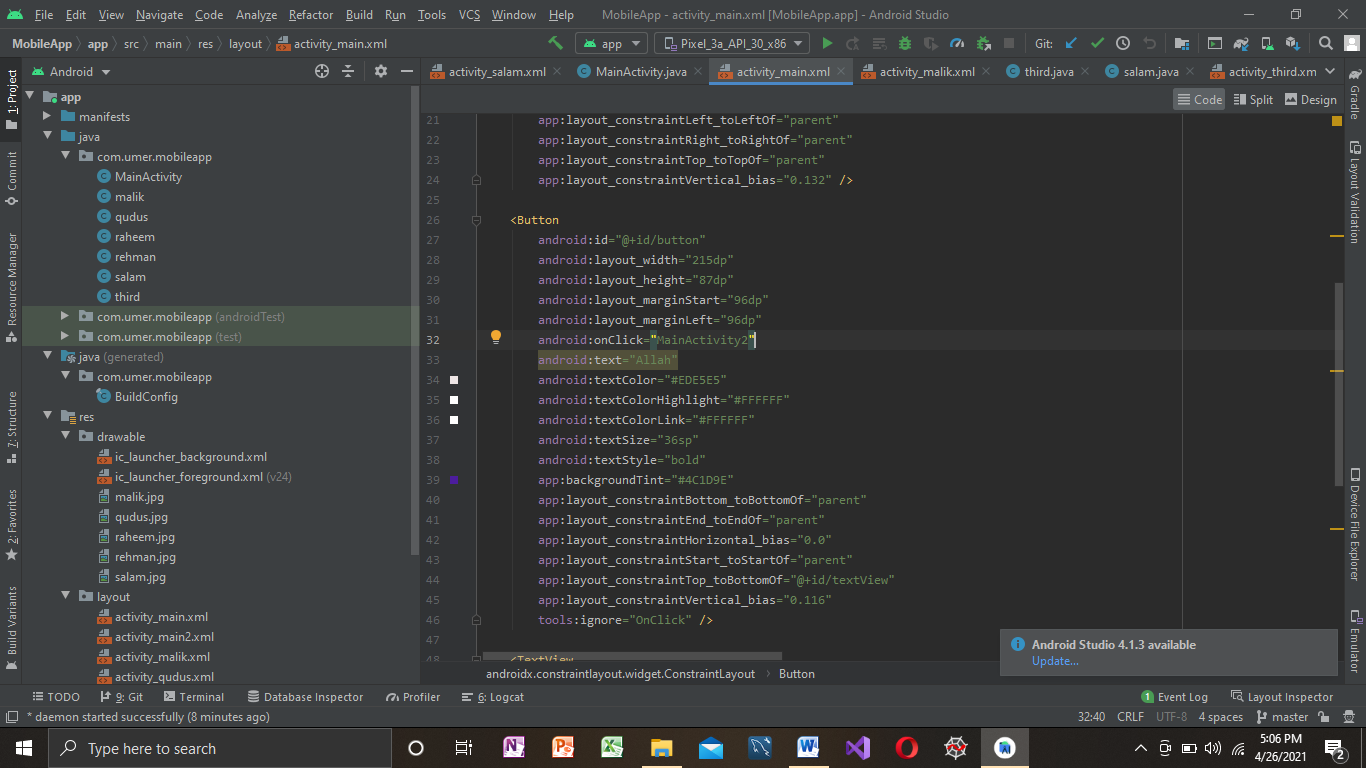


Constraints:

 A **constraint** represents a connection or alignment to another view, the parent layout, or an invisible guideline.

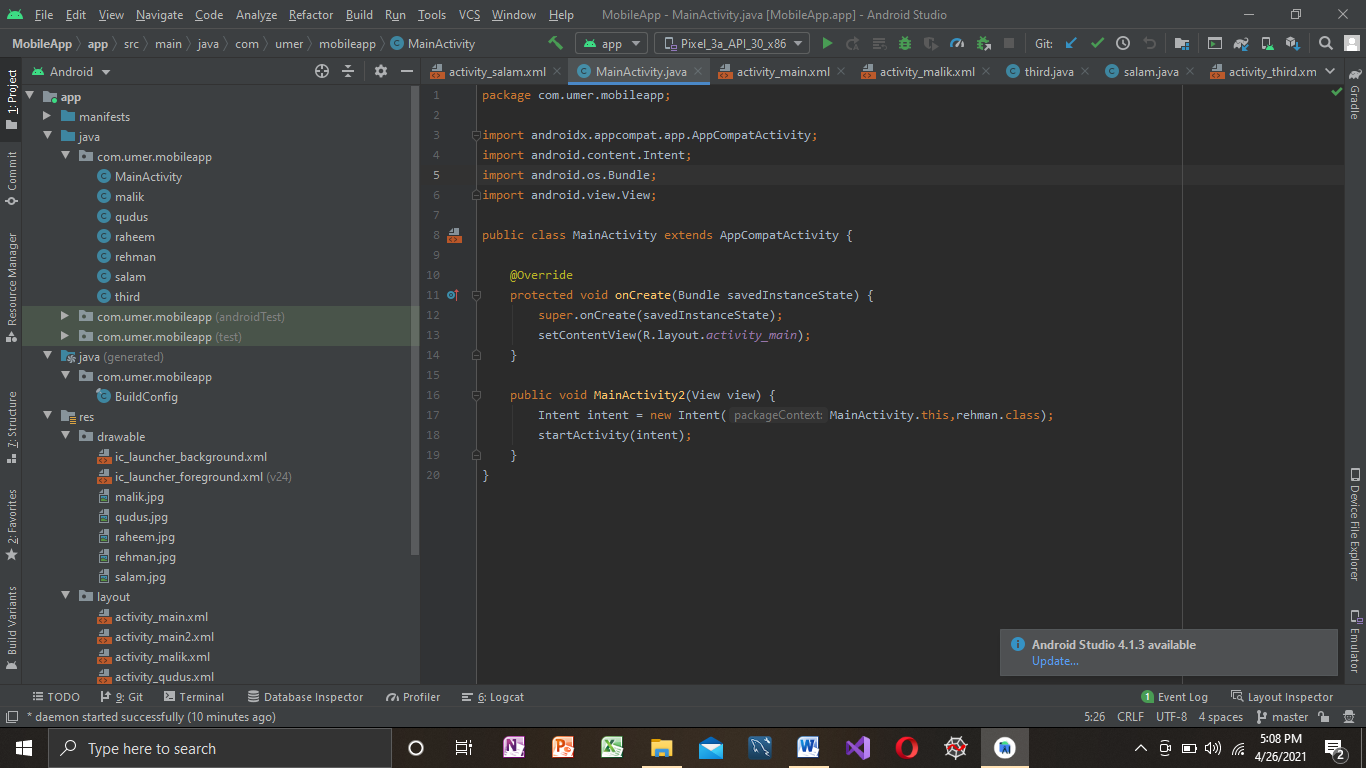


onClick Function:

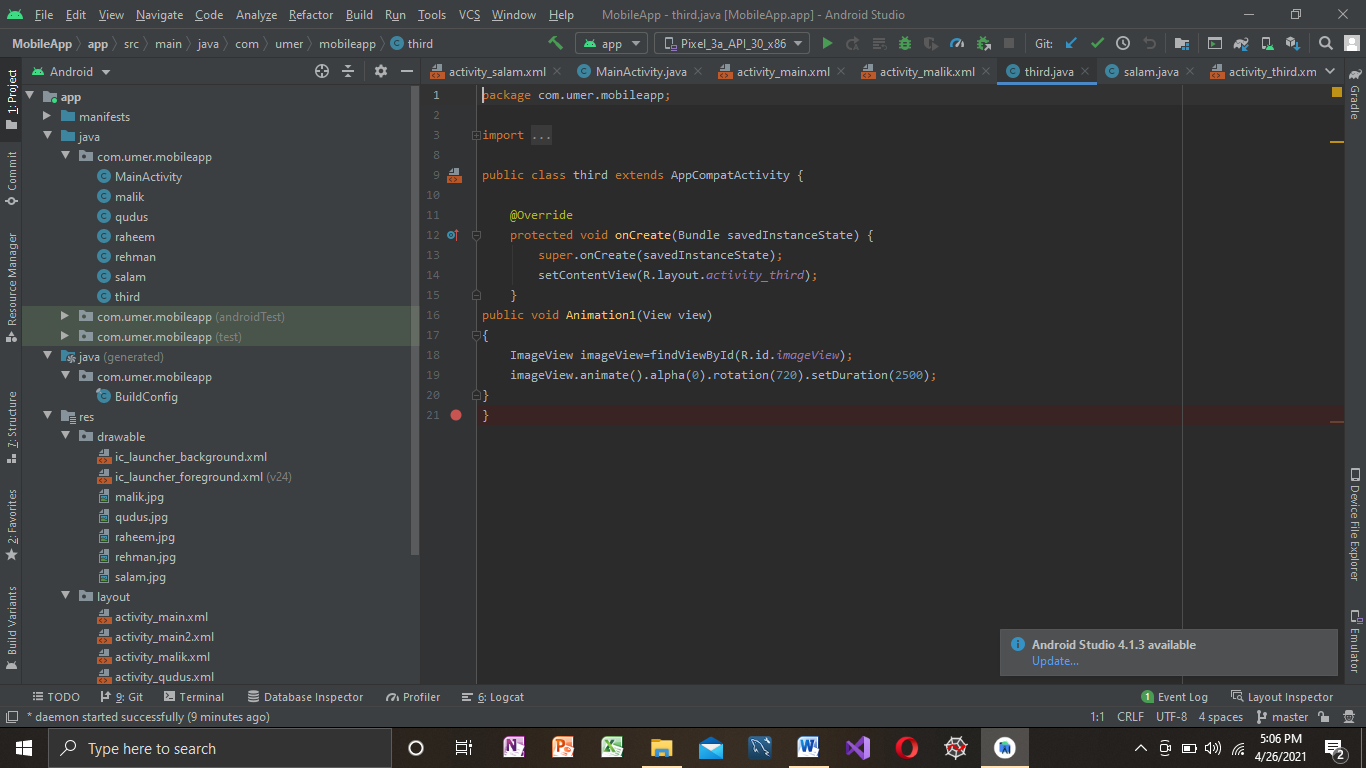


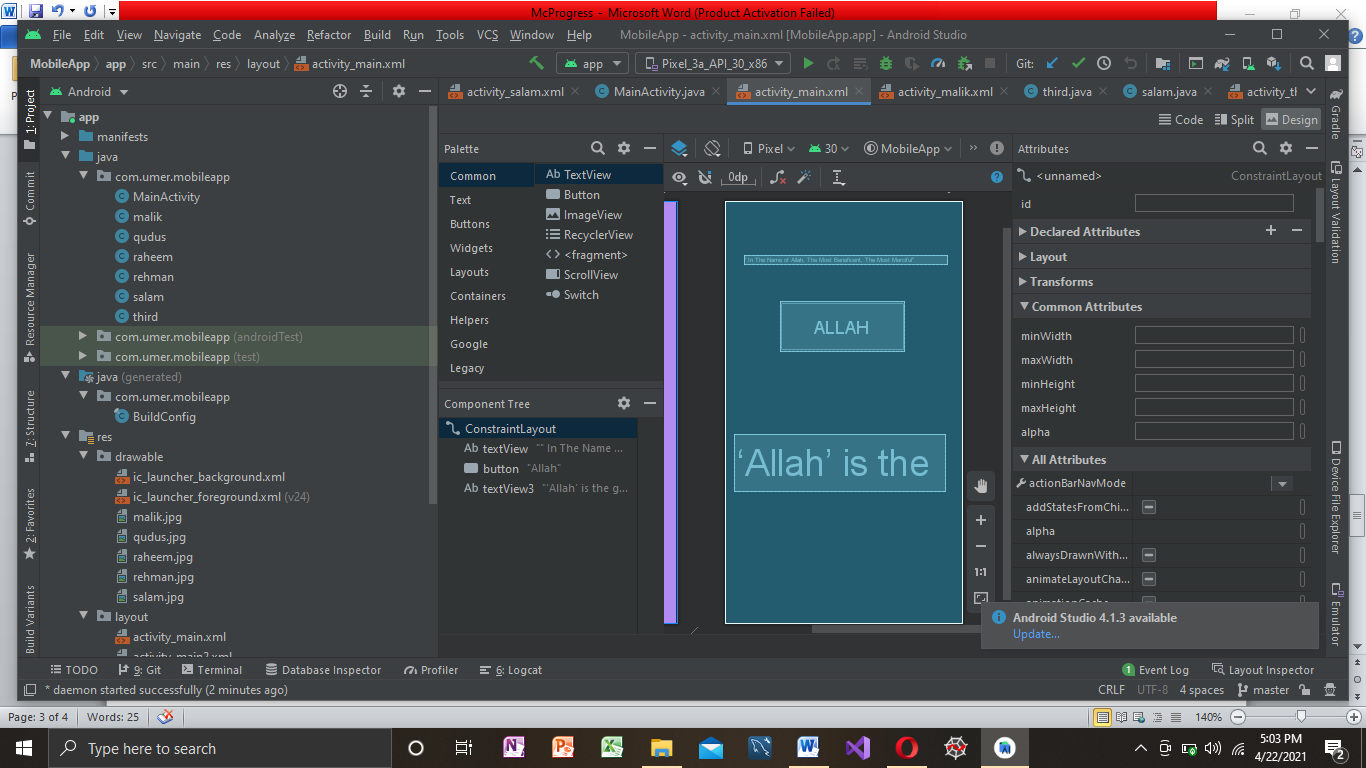
Intent:

**Android Intent** is the message that is passed between components such as activities, content providers, broadcast receivers, services etc. It is generally used with startActivity() method to invoke activity, broadcast receivers etc.



Animation:

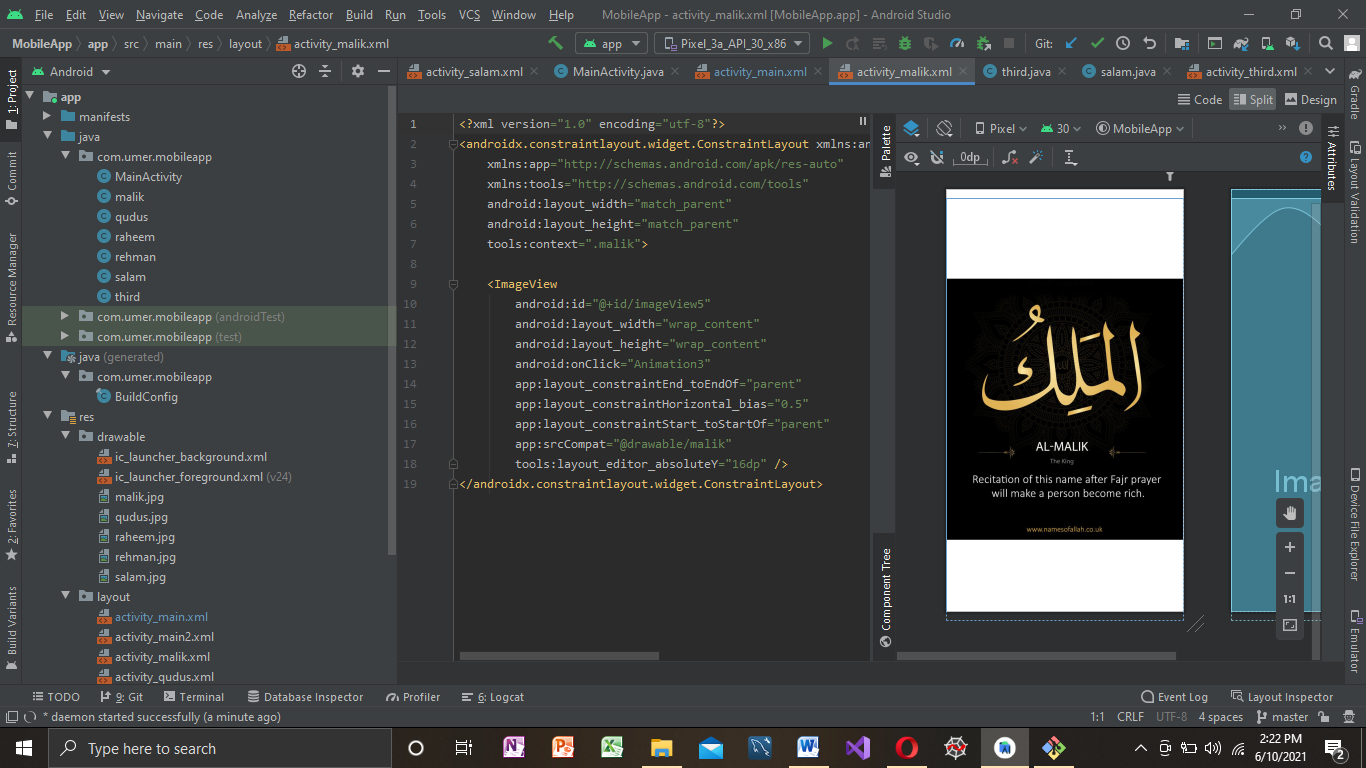


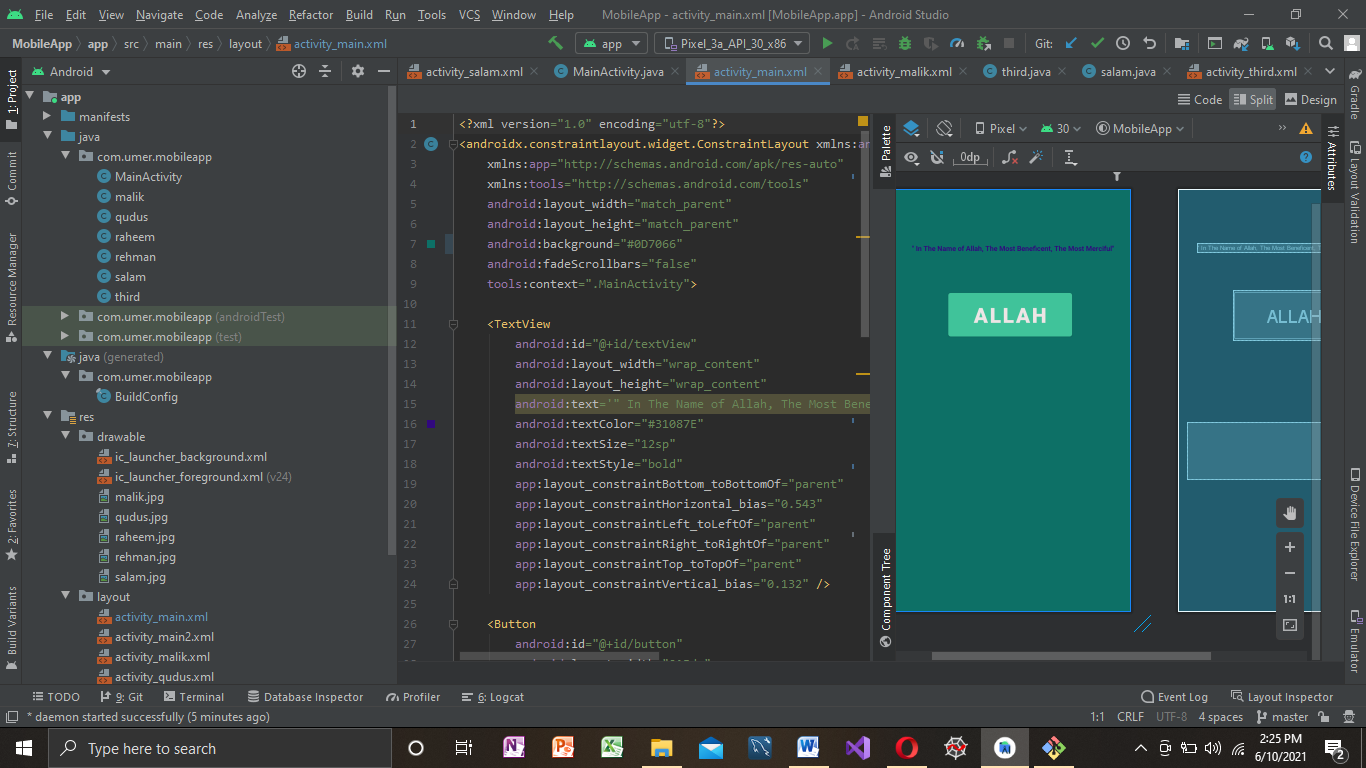


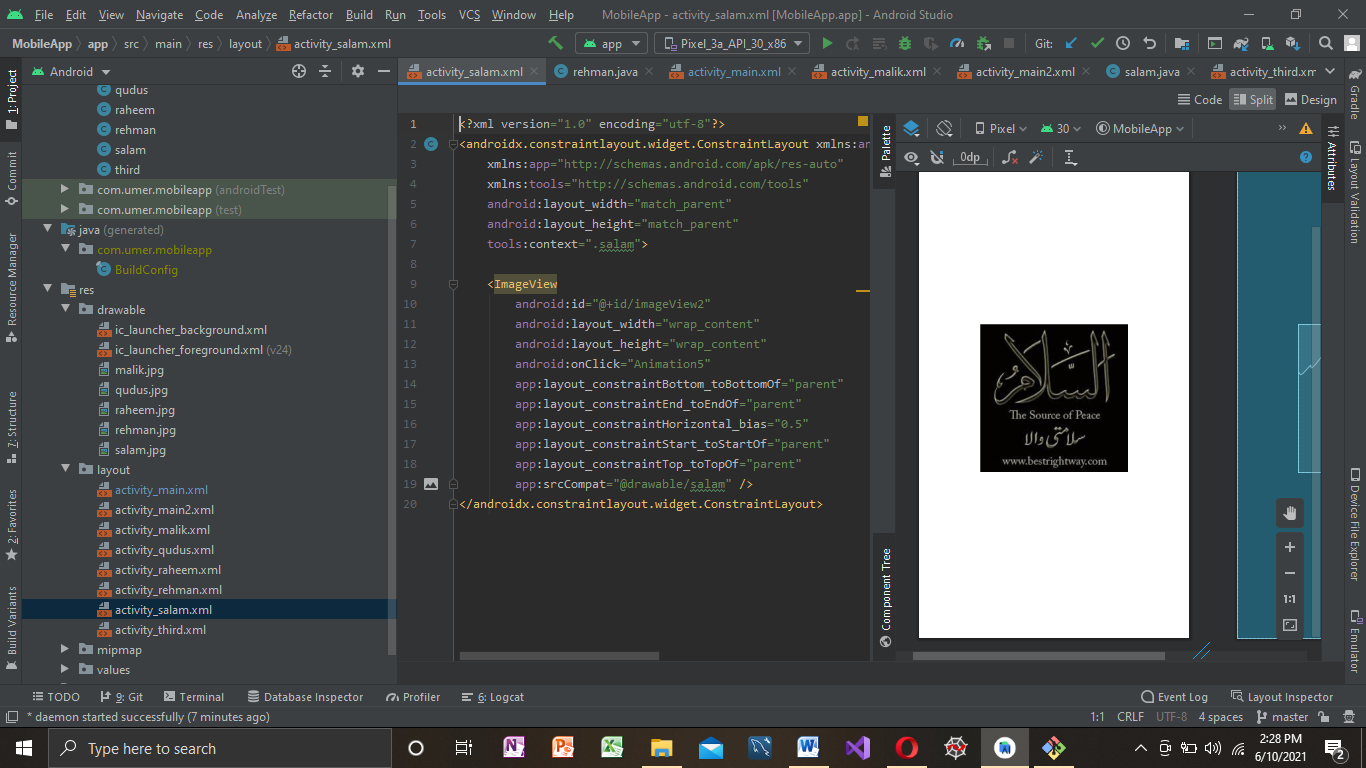
Basic Mobile Application:

 Video of basic app

Assignment 1:







DataBase:

A **database** is a collection of information that is organized so that it can be easily accessed, managed and updated. Computer **databases** typically contain aggregations of data records or files, containing information about sales transactions or interactions with specific customers.

## Define a schema and contract

public final class FeedReaderContract {  
    // To prevent someone from accidentally instantiating the contract class,  
    // make the constructor private.  
    private FeedReaderContract() {}  
  
    /\* Inner class that defines the table contents \*/  
    public static class FeedEntry implements BaseColumns {  
        public static final String TABLE\_NAME = "entry";  
        public static final String COLUMN\_NAME\_TITLE = "title";  
        public static final String COLUMN\_NAME\_SUBTITLE = "subtitle";  
    }  
}

## Create a database using an SQL helper

private static final String SQL\_CREATE\_ENTRIES =  
    "CREATE TABLE " + FeedEntry.TABLE\_NAME + " (" +  
    FeedEntry.\_ID + " INTEGER PRIMARY KEY," +  
    FeedEntry.COLUMN\_NAME\_TITLE + " TEXT," +  
    FeedEntry.COLUMN\_NAME\_SUBTITLE + " TEXT)";  
  
private static final String SQL\_DELETE\_ENTRIES =  
    "DROP TABLE IF EXISTS " + FeedEntry.TABLE\_NAME;

## Put information into a database

// Gets the data repository in write mode  
SQLiteDatabase db = dbHelper.getWritableDatabase();  
  
// Create a new map of values, where column names are the keys  
ContentValues values = new ContentValues();  
values.put(FeedEntry.COLUMN\_NAME\_TITLE, title);  
values.put(FeedEntry.COLUMN\_NAME\_SUBTITLE, subtitle);  
  
// Insert the new row, returning the primary key value of the new row  
long newRowId = db.insert(FeedEntry.TABLE\_NAME, null, values);

## Read information from a database

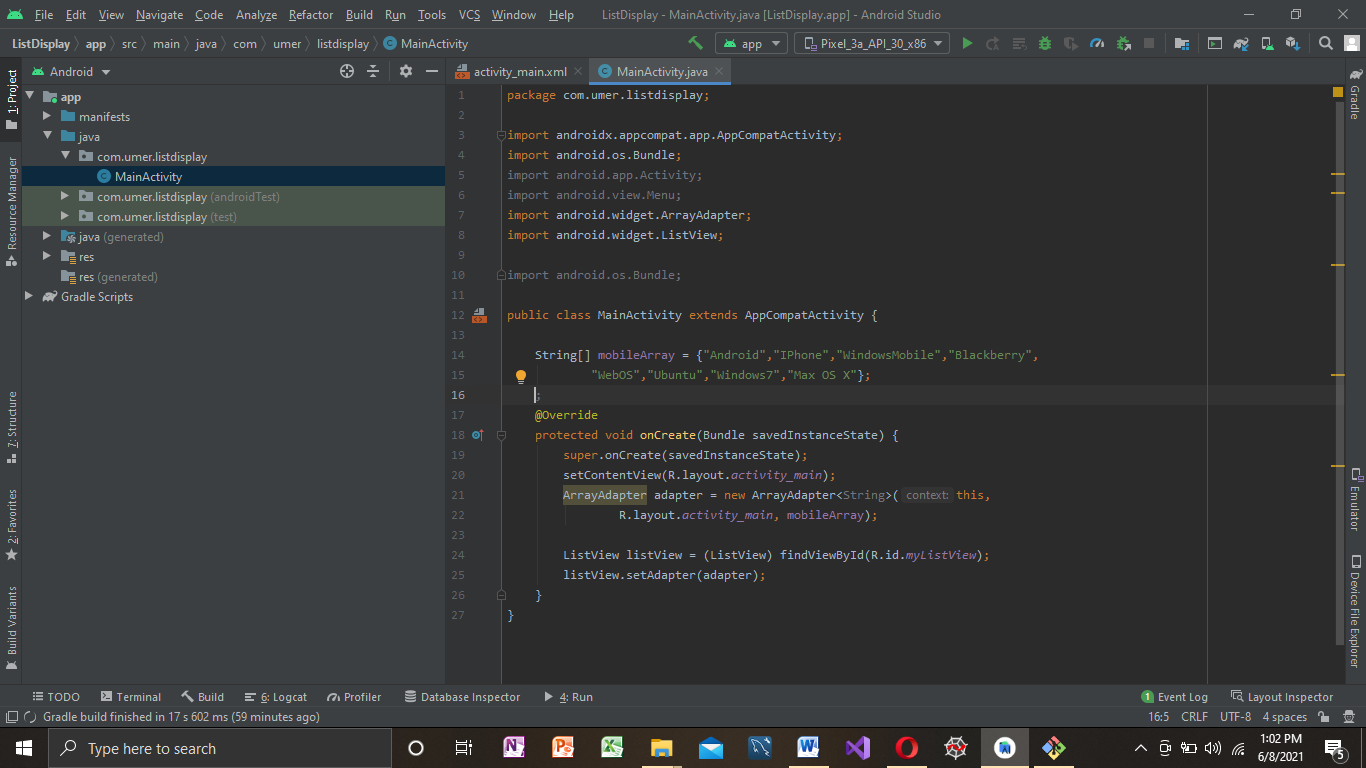
SQLiteDatabase db = dbHelper.getReadableDatabase();  
  
// Define a projection that specifies which columns from the database  
// you will actually use after this query.  
String[] projection = {  
    BaseColumns.\_ID,  
    FeedEntry.COLUMN\_NAME\_TITLE,  
    FeedEntry.COLUMN\_NAME\_SUBTITLE  
    };  
  
// Filter results WHERE "title" = 'My Title'  
String selection = FeedEntry.COLUMN\_NAME\_TITLE + " = ?";  
String[] selectionArgs = { "My Title" };  
  
// How you want the results sorted in the resulting Cursor  
String sortOrder =  
    FeedEntry.COLUMN\_NAME\_SUBTITLE + " DESC";  
  
Cursor cursor = db.query(  
    FeedEntry.TABLE\_NAME,   // The table to query  
    projection,             // The array of columns to return (pass null to get all)  
    selection,              // The columns for the WHERE clause  
    selectionArgs,          // The values for the WHERE clause  
    null,                   // don't group the rows  
    null,                   // don't filter by row groups  
    sortOrder               // The sort order  
    );

## Delete information from a database

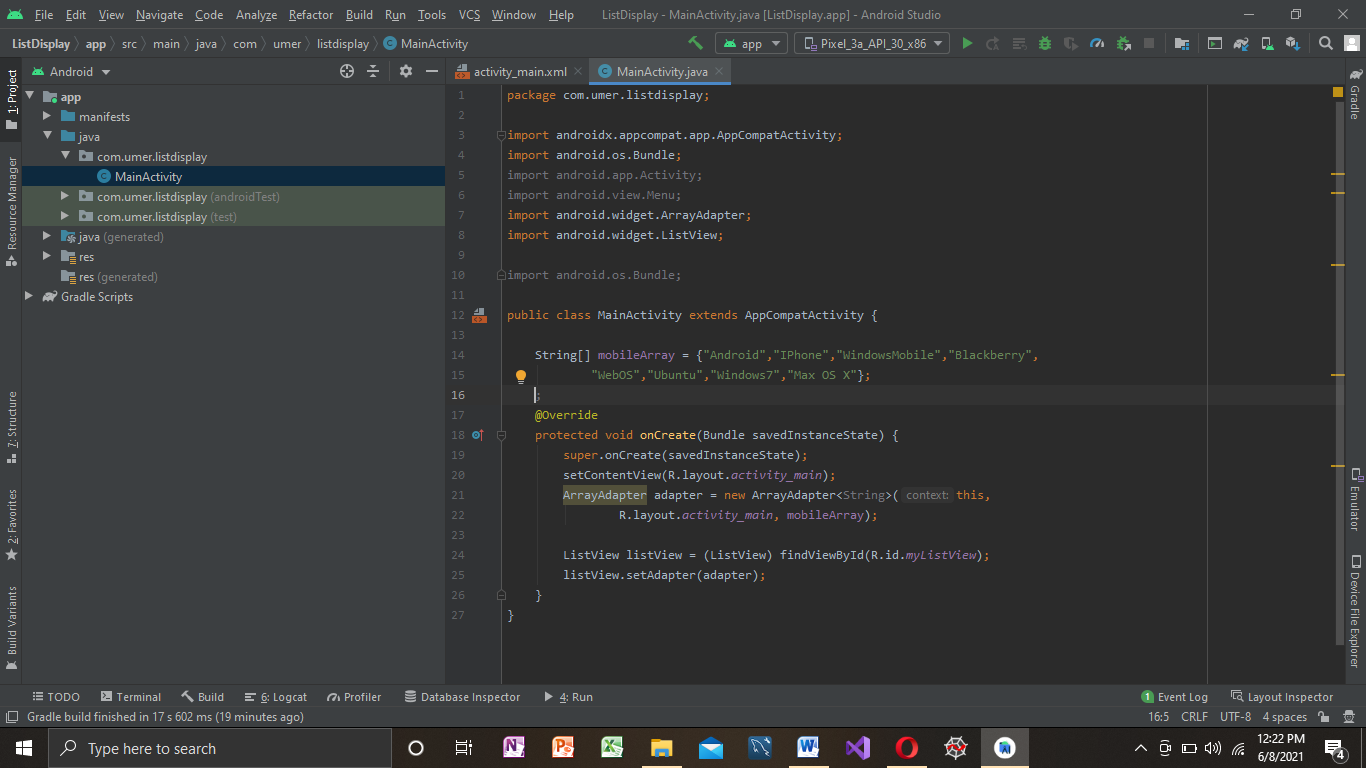
String selection = FeedEntry.COLUMN\_NAME\_TITLE + " LIKE ?";  
// Specify arguments in placeholder order.  
String[] selectionArgs = { "MyTitle" };  
// Issue SQL statement.  
int deletedRows = db.delete(FeedEntry.TABLE\_NAME, selection, selectionArgs);

Adapter:

Adapter is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and sends the data to an Adapter view then view can takes the data from the [adapter](https://abhiandroid.com/ui/adapter/) view and shows the data on different views like as list view



List View:

Android **ListView** is a view which groups several items and display them in vertical scrollable list. The list items are automatically inserted to the list using an **Adapter** that pulls content from a source such as an array or database. 



Drawer Navigation:

The **navigation drawer** is a UI panel that shows your app's main **navigation** menu. The **drawer** appears when the user touches the **drawer** icon in the app bar or when the user swipes a finger from the left edge of the screen.

