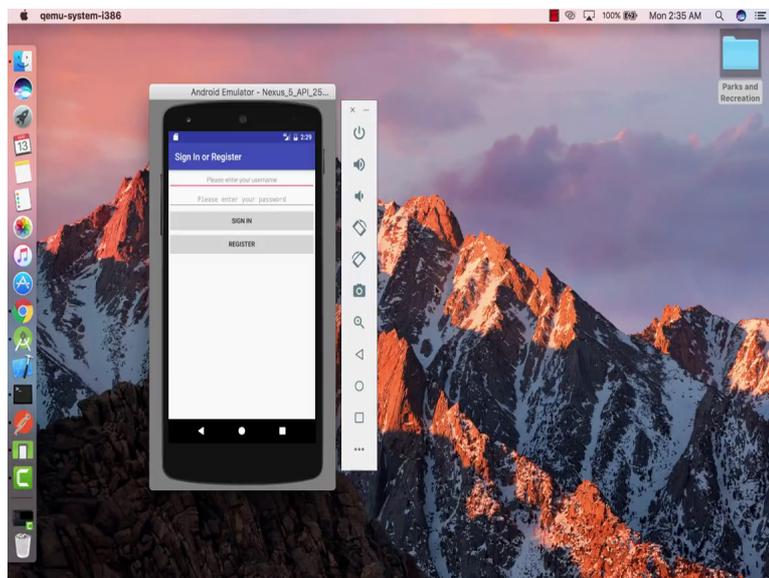


**Introduction to Morden Application Development**  
**Dr. Gaurav Raina**  
**Prof. Tanmai Gopal**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Module - 18**  
**Lecture - 33**  
**Deploying to Google Play Store**

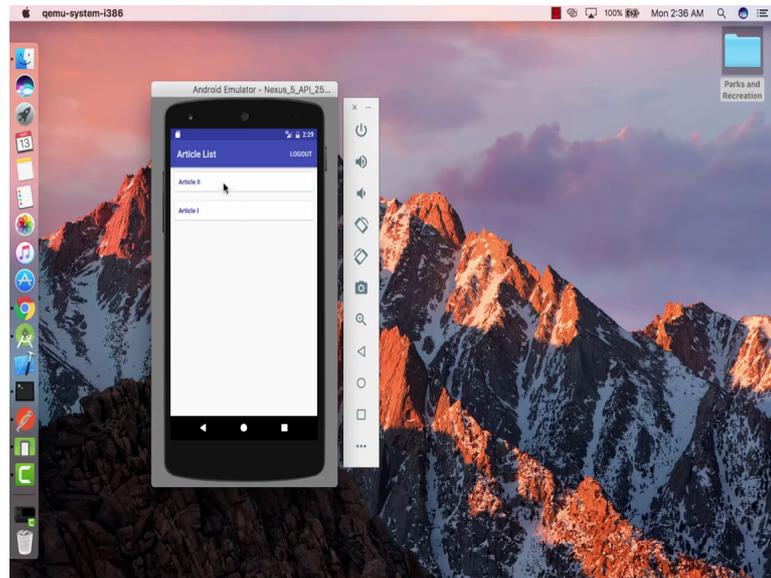
Welcome to module 18, in this module we are going to be deploying the app that we created in the previous module to the Google Play Store through the previous module we just created a simple blog app which should work as follows.

(Refer Slide Time: 00:11)



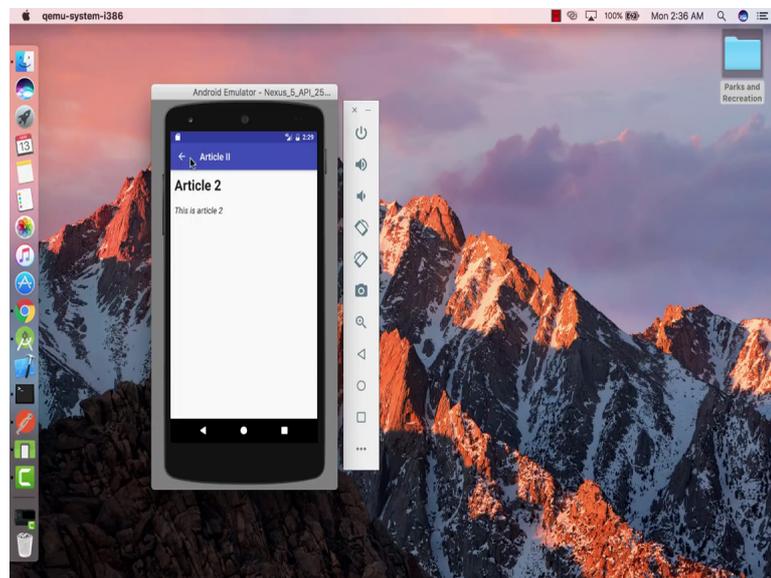
The first page is a authentication page wherein where presented is the option to sign in or register lets signing using our user id and password Imad and Imad.

(Refer Slide Time: 00:24)



The next page is an article list page after signing in we are presented with articles that we have in our database.

(Refer Slide Time: 00:33)



And clicking any of these articles we are taken to article detail page which has back button on clicking which we are taken to the previous page we have the heading of the article in the tool bar and then the detailed description of the article. We also have a logout button on the tool bar clicking on which we are taken back to the authentication page. Now let us publish this app to the Google Play Store.

(Refer Slide Time: 00:59)

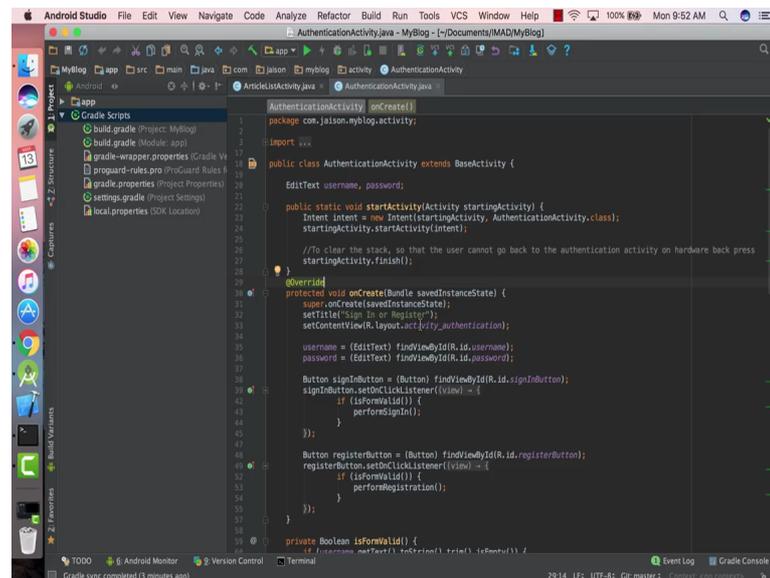
## Steps

- Create an account at the Google Play Developer console - <https://play.google.com/apps/publish>
- Pay the 25\$ registration fee.
- Generate a Signed APK - <https://developer.android.com/studio/publish/app-signing.html>
- Upload APK - Manage Releases
- Enter App details

Introduction to Modern Application Development Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

Publishing your app on the Google Play Store is a quite a straight forward process it starts by you creating an account at the Google play developer console and then paying the 25 dollar developer account registration fee and then we need to generate a signed APK for our for the app that we have just developed let us see how we can do that.

(Refer Slide Time: 01:17)



```
package com.jaison.myblog.activity;

import androidx.appcompat.app.AppCompatActivity;

public class AuthenticationActivity extends AppCompatActivity {

    EditText username, password;

    public static void startActivity(Activity startingActivity) {
        Intent intent = new Intent(startingActivity, AuthenticationActivity.class);
        startingActivity.startActivity(intent);
    }

    //To clear the stack, so that the user cannot go back to the authentication activity on hardware back press
    startActivity.finish();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_authentication);

    username = (EditText) findViewById(R.id.username);
    password = (EditText) findViewById(R.id.password);

    Button signInButton = (Button) findViewById(R.id.signInButton);
    signInButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (isFormValid()) {
                performSignIn();
            }
        }
    });

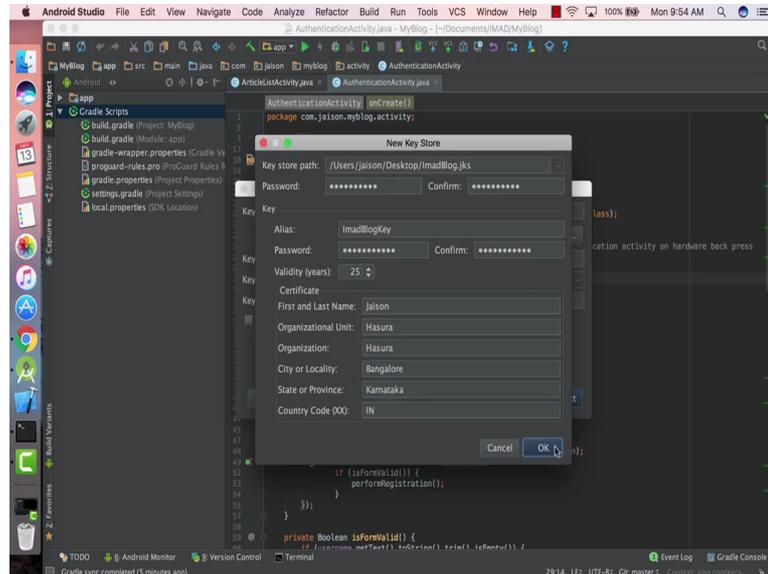
    Button registerButton = (Button) findViewById(R.id.registerButton);
    registerButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            performRegistration();
        }
    });
}

private Boolean isFormValid() {
    if (TextUtils.isEmpty(username.getText().toString()) || TextUtils.isEmpty(password.getText().toString())) {
        return false;
    }
    return true;
}
```

Signing the APK is important you basically create a public and private key and then use this public key certificate to sign your app what this public key certificate does is that it uniquely identifies you as the owner of this APK this ensures that nobody else can

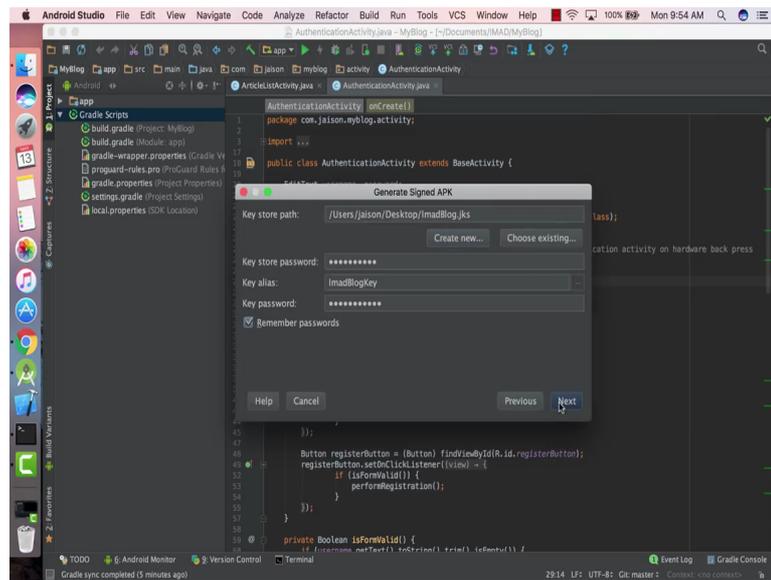
release and update your app on the Google Play Store. Now let us see how we can generate APK using android studio.

(Refer Slide Time: 01:42)

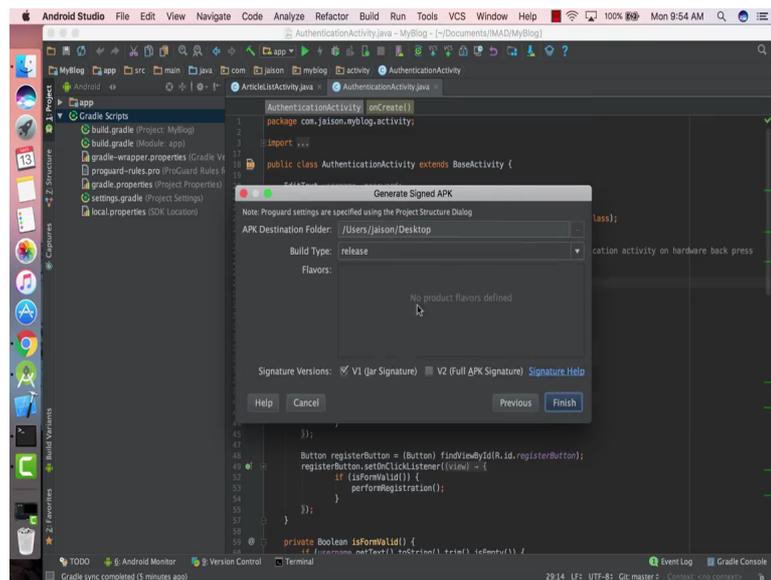


For that go to build click on generate signed APK next and now if you do not have a key store click on create new go to the path in which you want the key store to be generated in my case I am going to select the desktop and give a name to this key store call it Imad blog. This is the password to be used for the key store well tap anything that you choose, give the key store file an alias key and similarly give it a password this is the validity of this key store which is 25 years you can increase or decrease it to you are liking and then this is the normal information for the certificate that you are going to sign in your app this will not be shown anywhere else, but associated with your APK click on remember password.

(Refer Slide Time: 03:00)

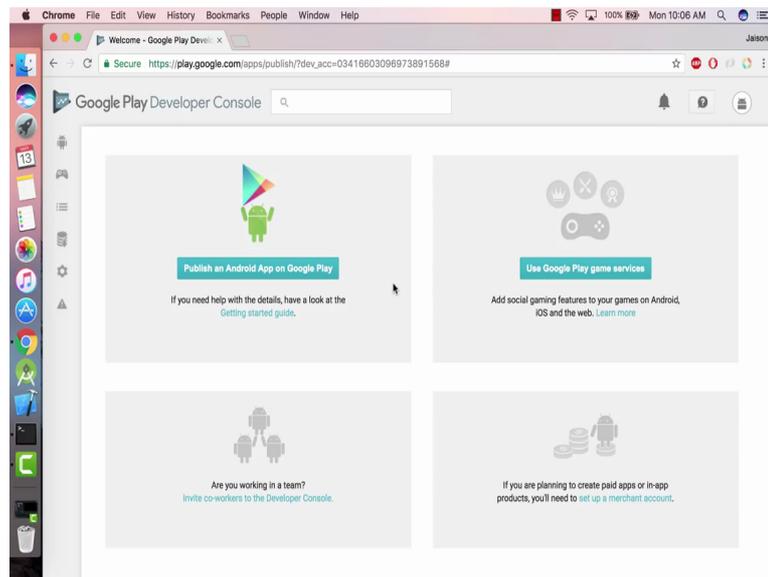


(Refer Slide Time: 03:03)



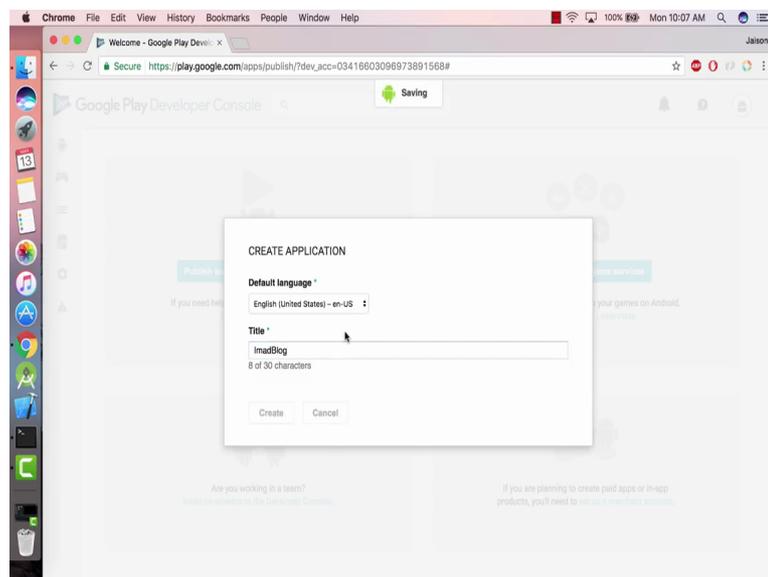
Click next check on V 1 jar signature and to read more about V 1 and V 2 click on signature help and you will be related to a link wherein more detailed information about this would be given.

(Refer Slide Time: 03:31)



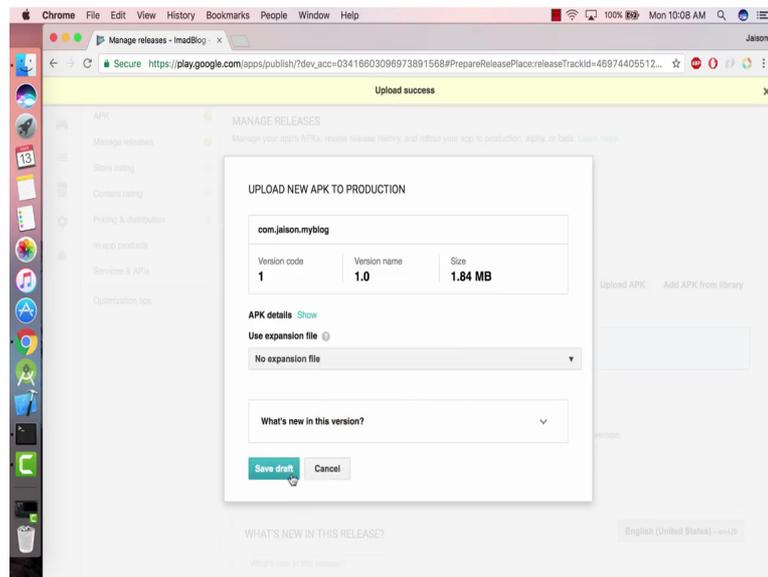
Major build type is set to release and click on finish. Once android studio has generated the signed APK keep a note on where APK is saved and now let go on to the Google developer console, if you registered properly and paid the developer fee you will be presented with this page.

(Refer Slide Time: 03:36)



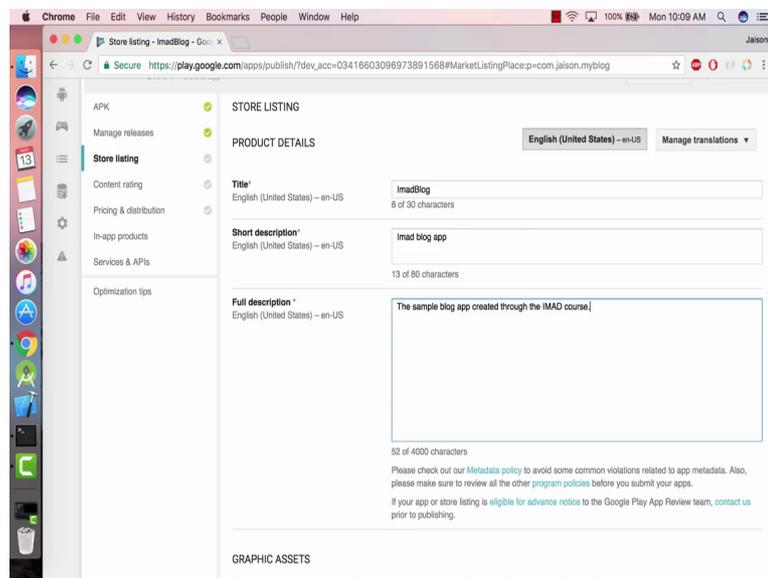
Click on publish an android app on Google play let the default language be English and give the title as Imad let it be Imad blog click on create.

(Refer Slide Time: 03:51)



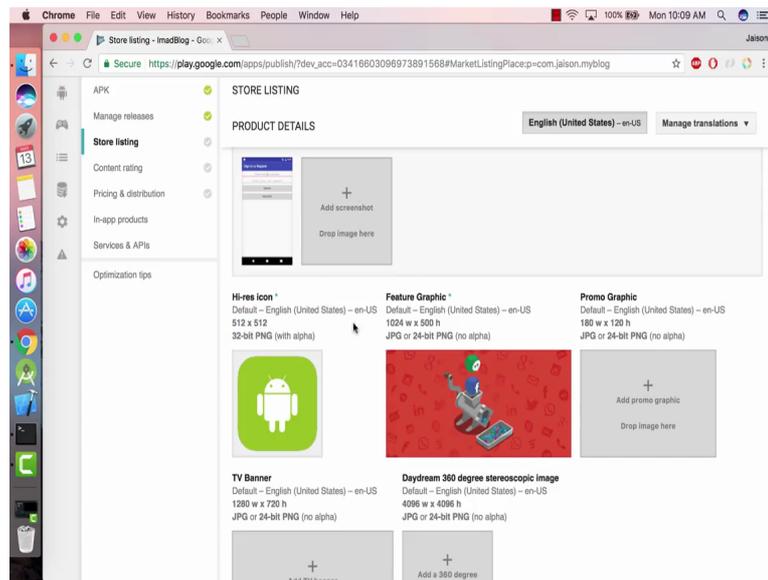
Now, first thing go to manage releases scroll down to manage production click on create release and click on upload APK browse files and select an APK let the APK upload not that is done click on save draft.

(Refer Slide Time: 04:30)



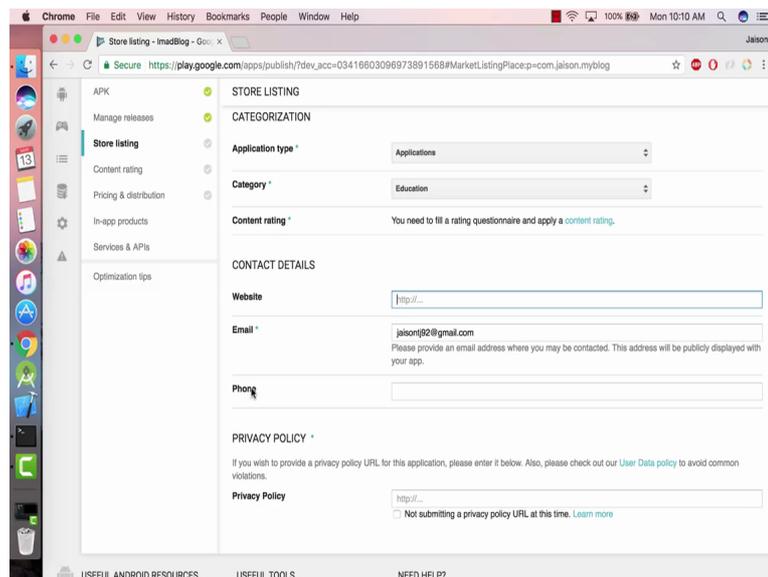
Next thing to do is go to store listing one second click on save; give a short description about the app give a full description.

(Refer Slide Time: 04:50)



Next thing is to submit screen shots and icons for your app I already have a few screenshots and random icons. So, I right click for this, the thing to keep in mind is that the higher is icon and the feature graphic are mandatory fields and they have a specific size and type associated with them. So, you have to make sure that the image that you upload is of this type.

(Refer Slide Time: 05:21)

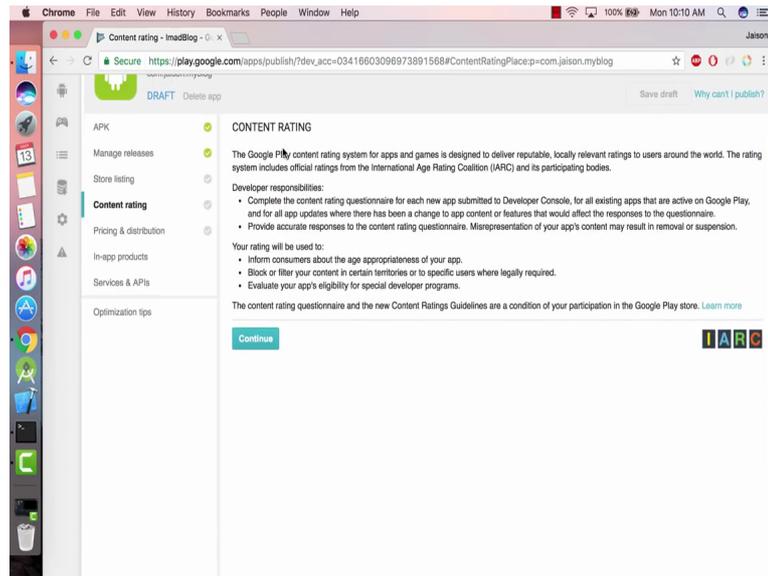


Finally select an application type whether it is an application or a game I will select as application and select a category for the app I would just select education and then

provide your contact detail if you have a website provide the website link if you have a phone number, give your phone number and email id is a mandatory field.

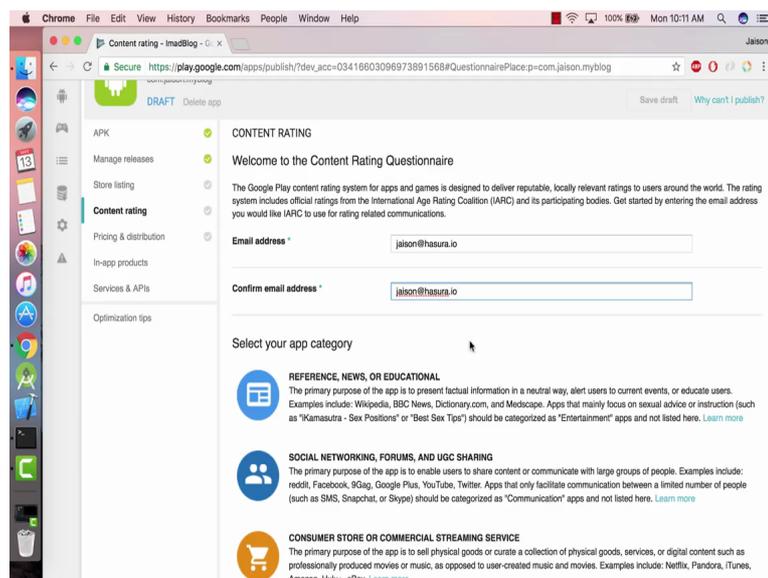
Similarly, for privacy policy you can have a website which shows your privacy policy or for now let us say not submitting a privacy policy and let us save this.

(Refer Slide Time: 05:56)



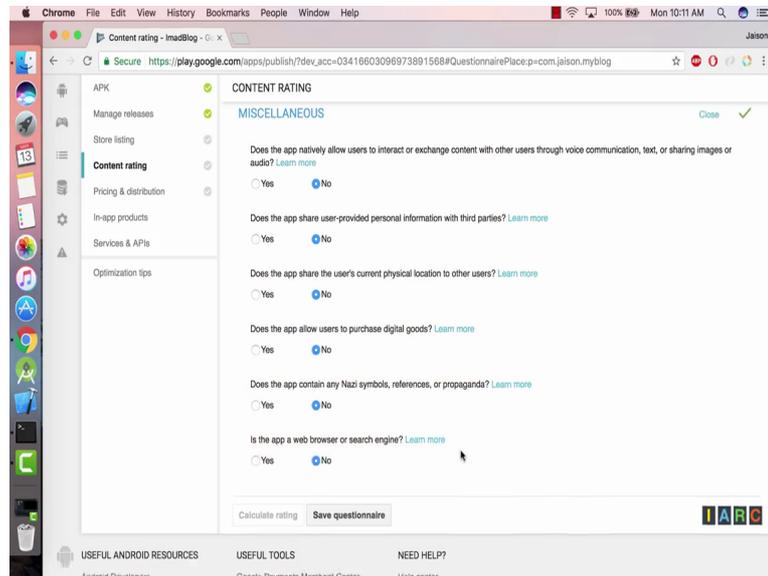
This is the content rating that we have to provide for your app provides an email address.

(Refer Slide Time: 06:00)



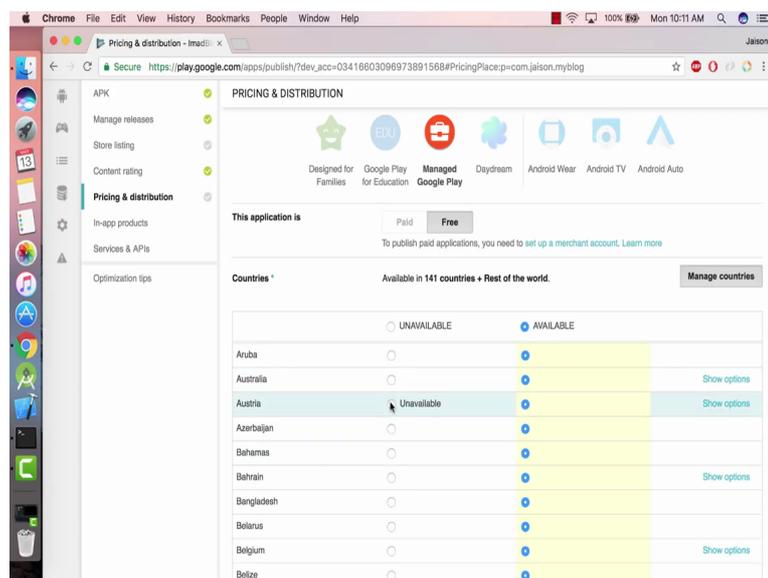
Confirm that email address and select your app category. So, I am going to say utilities violent material no sexual material or nudity no potentially offensive language no controlled substance no.

(Refer Slide Time: 06:14)



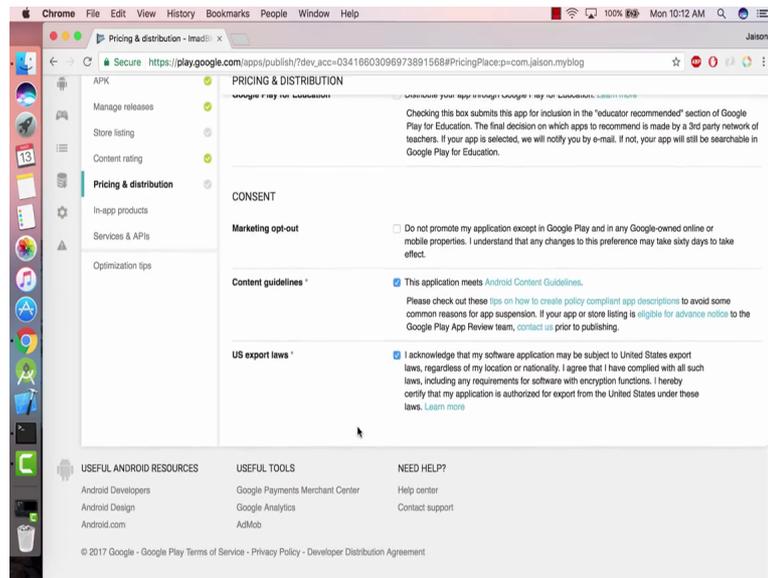
So, basically just read through all of these and select the appropriate answers for that relate your app in my case everything is no and then apply it basically this is the same that (Refer Time: 06:46) for different cases, alright.

(Refer Slide Time: 06:52)



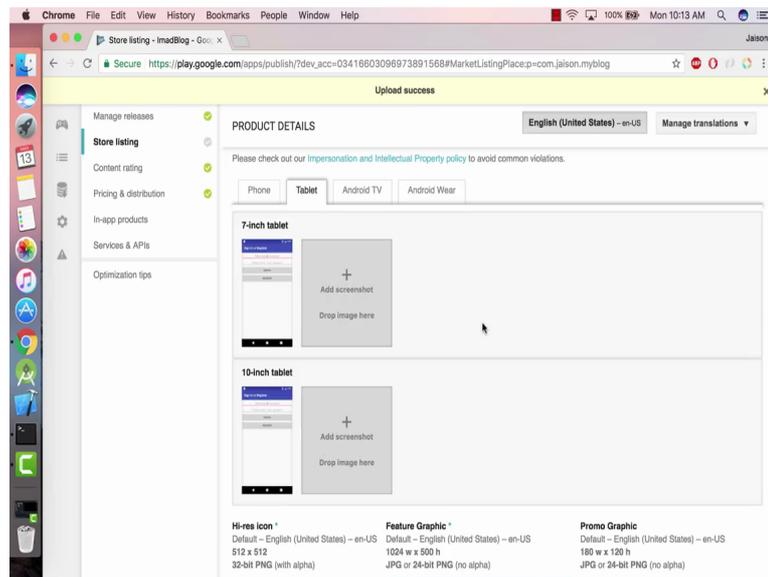
Let us go to price and distribution select the free tier make it available for every country or you can individually select the countries you wanted to be unavailable for and you can select whether you are showing ads in your app we are not, the device categories that will support none of these Google play for education we do not want that.

(Refer Slide Time: 07:24)



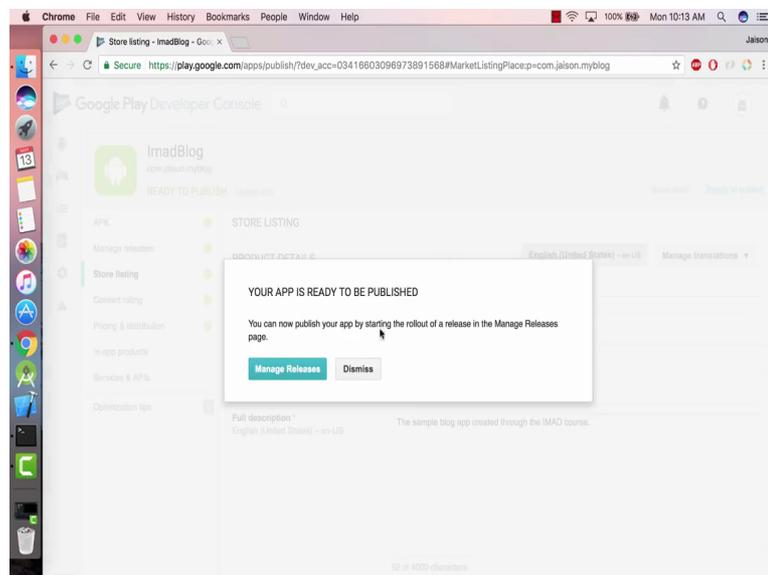
This is asking whether you do not want Google play to promote your app that is not true content guidelines it meet it US export laws true after you have done that save the draft. Now you still do not have the publish icon. So, let us see why that is, it is saying that you need to add at least 2 non android TV screenshots. Let us go to our store listing and see where are those.

(Refer Slide Time: 07:58)



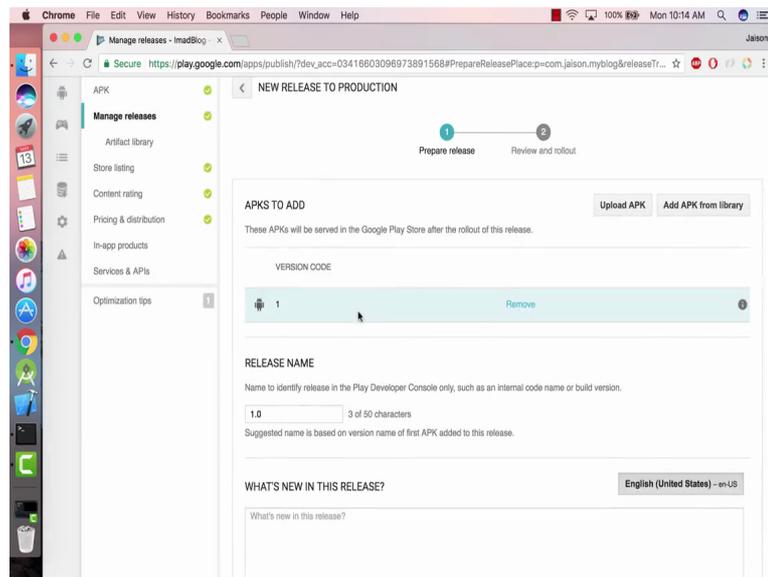
So, we need to select listing we need to have add a screenshot for our tablet and at 7 inch and a 10 inch tablet. So, I am going to add the same screenshot in this case you are supposed to turn your app on your 7 inch and 10 inch tablet and then take the screenshots.

(Refer Slide Time: 08:28)

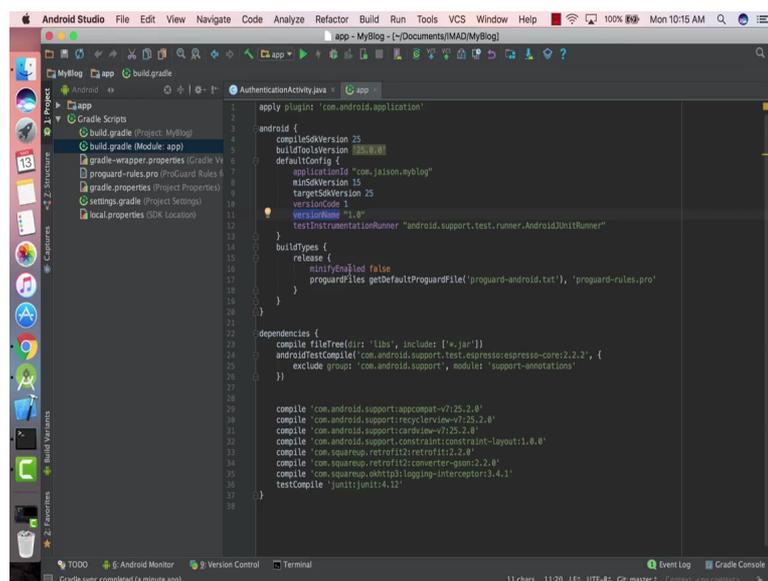


And once you are done everything right you will get this button which says ready to publish click on ready to publish and click on manage releases. Let us see what this version code and release name is let us go to android app go to build dot Gradle file.

(Refer Slide Time: 08:42)

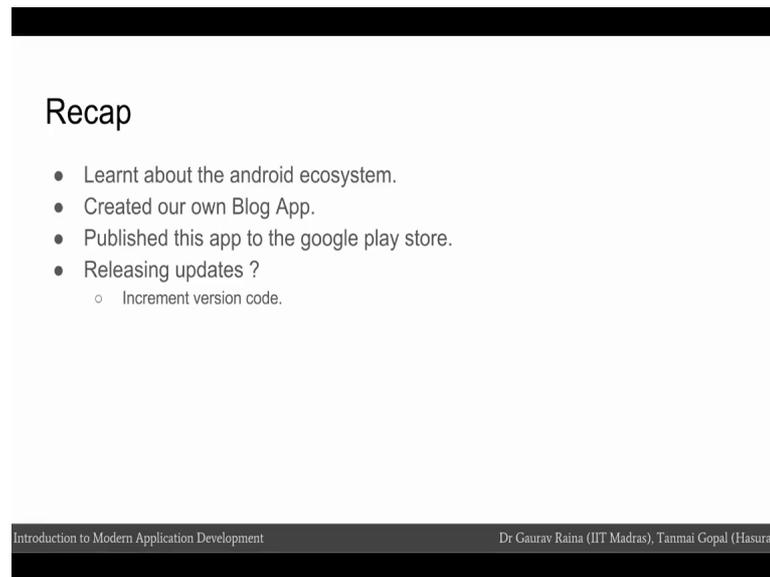


(Refer Slide Time: 08:47)



See there is a version code and version name provided here for every update you release you have to increment this version code it has to be greater than the previous version code and the version name can remain the same this is just a track of the version name of the app on the play store and click on review and just click on start rollout to production click on confirm your app will get live on the Google Play Store in a few hours. All right and that is it.

(Refer Slide Time: 09:33)



## Recap

- Learnt about the android ecosystem.
- Created our own Blog App.
- Published this app to the google play store.
- Releasing updates ?
  - Increment version code.

Introduction to Modern Application Development Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

This is how you release an app to the Google Play Store with this we come to the end of the android application development series. So, in today's module and in the series we learnt about the android ecosystem we created our own blog app and we published this bog blog app to Google Play Store what remains is to know how to release updates releasing updates is also quite straight forward and it is a simply crosses you simply have to sign into your Google developer console and go to manage releases and create and uploading new APK.

The only thing you have to make sure of is that you have to increase the version code by one and also sign the APK with the same key store file that you created while uploading the first as PK. It is very important that you keep your key store file in a very safe place and do not share with anyone or do not lose it because if you lose your key store file then you will not be able to upload anymore updates to your apps and if somebody else gets your key store files then he will be easily able to release updates in your name.