ICP PRESENTATION TWO

GROUP – 17

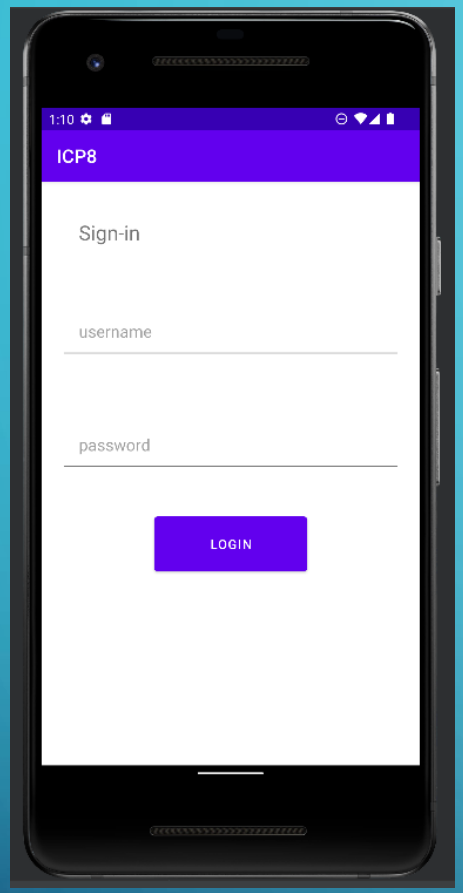
MANIKANTA KAVITAPU

KAVYA PARA

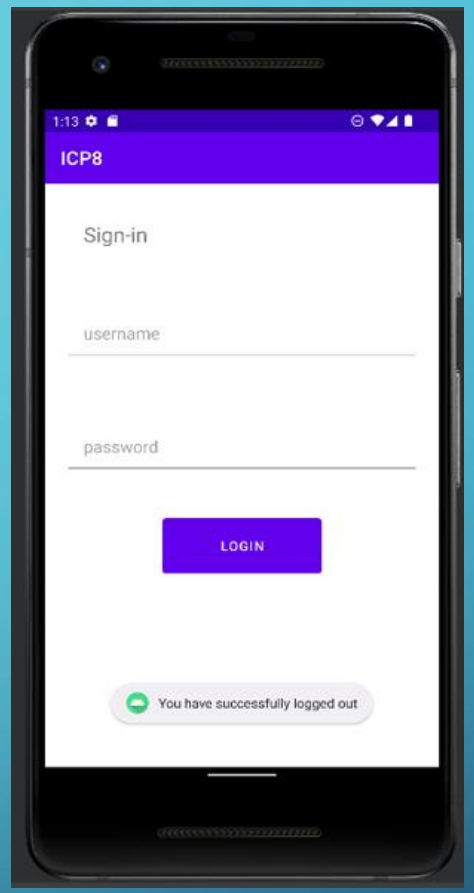
ICP8:

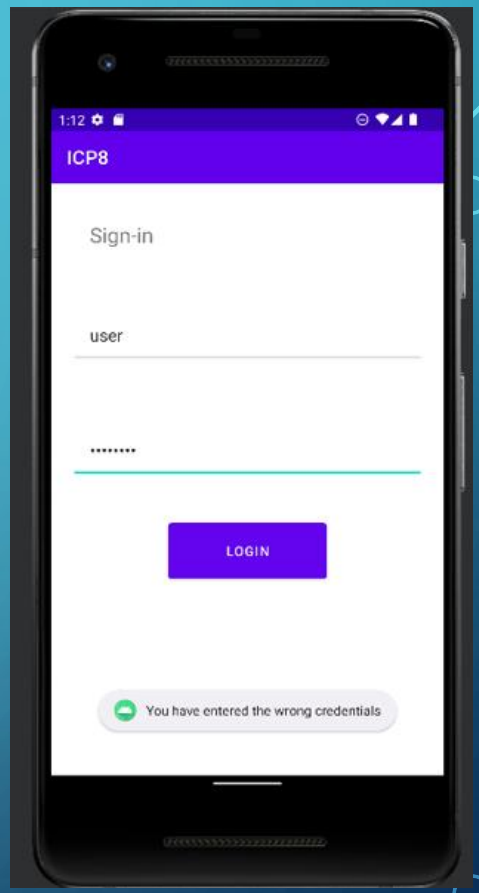
- ICP8 can be basically considered as an introduction to Mobile application and an Android Studio.
- In this ICP, we have installed the Android Studio and created a simple sign in page which after logging in navigates to a welcome screen.
- We have used two simple editTexts and a button for the main screen.
- By using the Toast we have generated a alert message if the credentials are wrong.

```
/*Displaying The Appropriate Message If The Credentials Are Invalid*/
else{
    Toast.makeText( context: MainActivity.this,  text: "You have entered the wrong credentials",Toast.LENGTH_SHORT).show();
}
```


1:10
ICP8
Sign-in
username
password
LOGIN

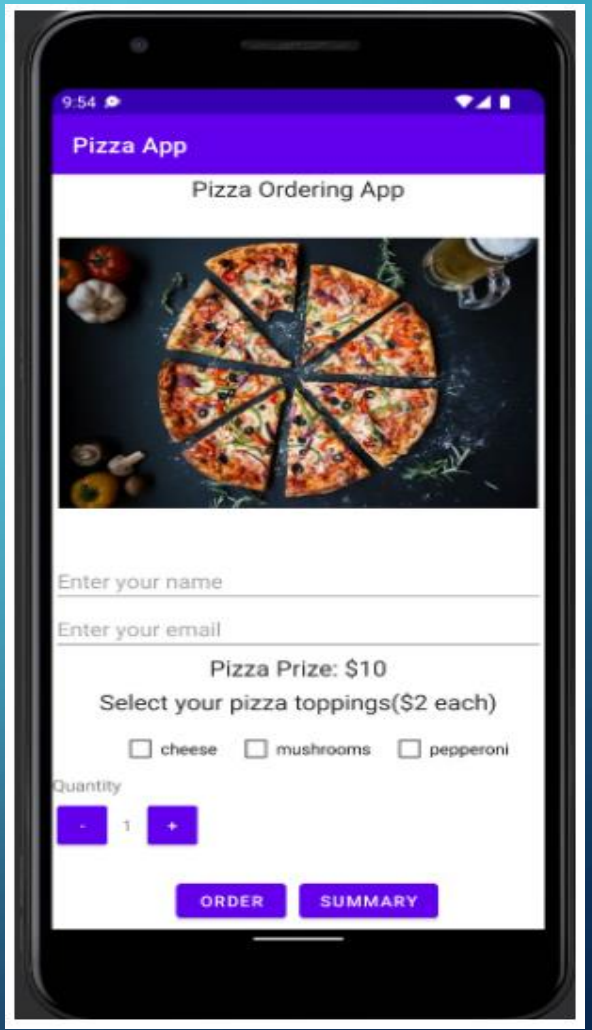

1:11
ICP8
Welcome to ICP8 !!!
LOG OUT


1:13
ICP8
Sign-in
username
password
LOGIN
You have successfully logged out

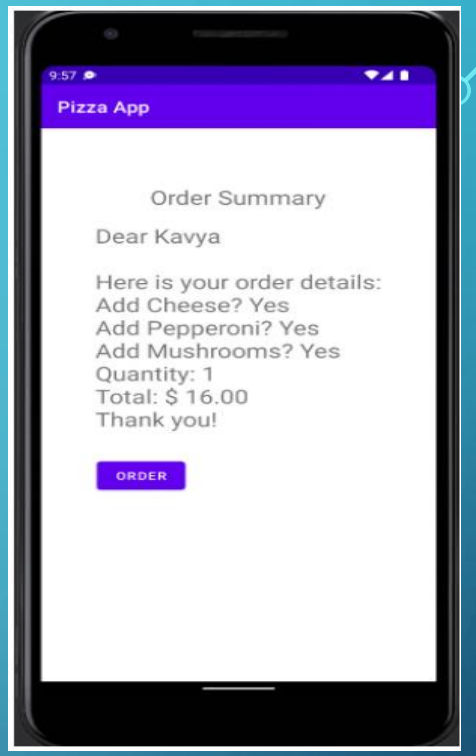

1:12
ICP8
Sign-in
user
LOGIN
You have entered the wrong credentials

ICP9:

- ICP9 is about developing a pizza ordering application using Android Studio.
- We have developed a basic pizza ordering app where the user can have some toppings to opt from, the price details and the quantity.
- The user also has to provide basic details like name and email so that the order summary can be mailed to user's email.
- We have used two buttons in the main screen: order and summary.
- After filling the details and selecting the toppings, quantity, whenever user presses the summary button it will navigate to summary page where the user can verify the entire order.



- The summary page contains all the order details and also a button which will help the user to navigate to the main screen.
- We have used the intent for these screen navigations.




```
public void onClick(View v) {
    Intent intent = new Intent( packageContext: Summary.this, MainActivity.class);
    startActivity(intent);
}
```

ICP10:

- In this ICP, we have fetched the user details from the web using Retrofit.
- Retrofit is basically is a type-safe HTTP client which is used for Java and Android. It will turn our HTTP API into a Java interface. It will allow us to retrieve and upload JSON using a REST based.

```
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://api.github.com/")
        .addConverterFactory(GsonConverterFactory.create())
        .build();
```

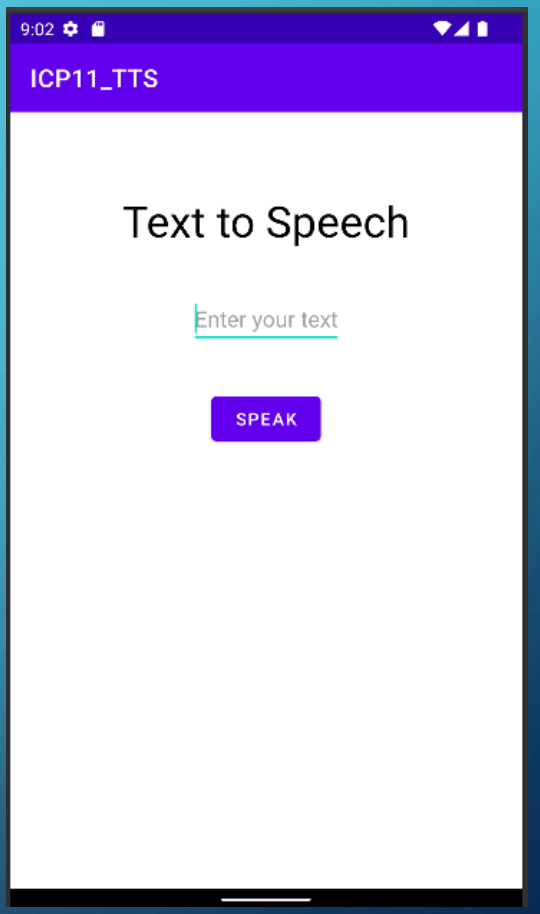
- As we are fetching the user details from the web, which makes our application to use the internet we have to add an extra permission in our AndroidManifest.xml file. If we don't add this permission in the xml file, we will not be able to use the internet.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

ICP11:

- ICP11 is about the Text to Speech.
- In this ICP, we have took the user input text and converted it into speech.
- We have used a simple editText for the user to enter the input and a button to convert the text into speech in the main screen.
- For the input text to be converted into speech, we have imported “android.speech.tts.TextToSpeech” and created a function called speak()

```
void speak(){
        String text = editText.getText().toString();
        tts.setSpeechRate(0.5f);
        tts.speak(text, TextToSpeech.QUEUE_ADD, params: null);
    }
```



- Here QUEUE_ADD is where the new entry of the text is added to the end of the playback queue in queue mode.
- We can also use QUEUE_FLUSH if we want to drop all the entries in the playback queue and replace them with new entries.
- The stop() will interrupt the current speech and discards all the remaining text too.
- The shutdown() will release all the resources.

```
@Override
protected void onPause(){
    super.onPause();
    tts.stop();
}
@Override
protected void onDestroy(){
    super.onDestroy();
    tts.shutdown();
}
```

Thank You