**WEB AND MOBILE PROGRAMMING**

**Group: 07**

**Students:**

**Venkata Naga Sai Srikanth Gollapudi, sggc9@umsystem.edu**

**Varun Reddy Mangi, vmf9@umsystem.edu**

**Video link:** <https://youtu.be/TuVsdzCnGH0>

**PPT link:**

<https://github.com/GollapudiSrikanth/WebMobileSpring2022/tree/main/Mobile/ICP_Presentation>

**ICP-8: Android Studio Login Page**

1. Created a project and installed Google pixel 5 with Android OS version of 12.

2. Now, we have created a interface for login page which contains Username and password fields and also a login button.

3. The above fields are created using EditText option with type as user and password.

4. Login button is created using button option.

5. All the business logic is witten in the MainActivity.java.

6. On entering valid details and clicking on login button, user will be directed to the home page.

7. The home page contains only a logout button. If the user clicks the logout button, then the user will be directed to the login page.

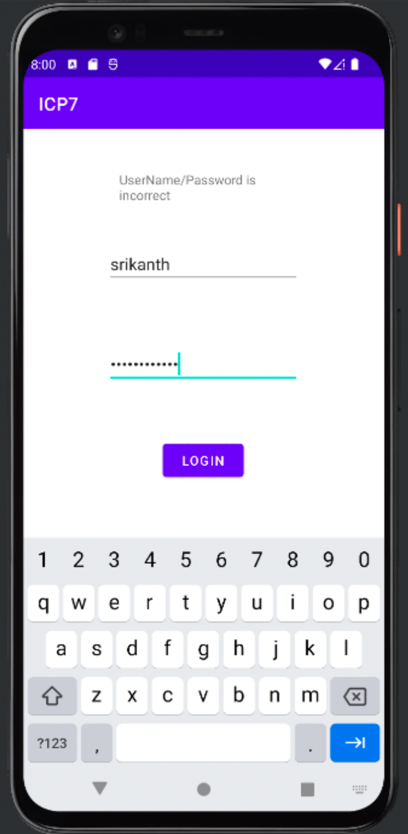
8. If user enters invalid details, an error message called “Username/Password is incorrect” will be displayed on the screen.

From this ICP we are able to implement a design from UI Wire Frames.

**OUTPUT:**

**A close-up of a cell phone

Description automatically generated with medium confidence A picture containing text, monitor, screenshot, electronics

Description automatically generated **

**ICP-9: API CALLING**

We have created a pizza ordering application using android studio.

1. This app basically helps users to order pizzas of their liking.

2. The interface contains Text edits which asks details from the user (user name and the user email id).

3. We have added 4 different toppings from which a user can select any by just clicking on the check box.

4. User can increase or decrease the quantity of the toppings by just clicking on + or – buttons.

5. We have also included summary and place order buttons.

6. On clicking summary button, the application displays the order summary.

7. On selecting place order, the application sends a summary mail to the respective email id.

From this ICP we are able to create and Pizza application and able to order a pizza with report sent to customer email ID.

**OUTPUT:**

A screenshot of a phone

Description automatically generated with low confidence A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a cell phone

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated

Summary page logic in the PizzaApp

Text

Description automatically generated

Order page design code in the PizzaApp adding scroll bar to the application

Text, letter

Description automatically generated

**ICP-10: Consuming the Rest Api**

1. Created the project using Android studio and added two dependencies which are retrofit2.9.0, converter-gson:2.9.0.

2. Created a java class file “User.java” which acts as a data transfer object which consists of id and username with getters and setters.

3. In the layout added a text field with id of ‘text1’ and added a nested scroll view to the

text1 which gives the way to scroll up and down.

4. In the onCreate function in the MainActivity.java,retrofit api is used consumed the

github rest api to fetch the users data.

5. On Successful call of api, response is collected in the onResponse method and casted to

User class which is collected in the list.

6. The collected list is then displayed in the text view which is present in the

MainAcitivity.xml and displayed to the user.

From this ICP we are able to consume rest api in the mobile application and got to know the dependencies of it

**OUTPUT:**

A screenshot of a phone

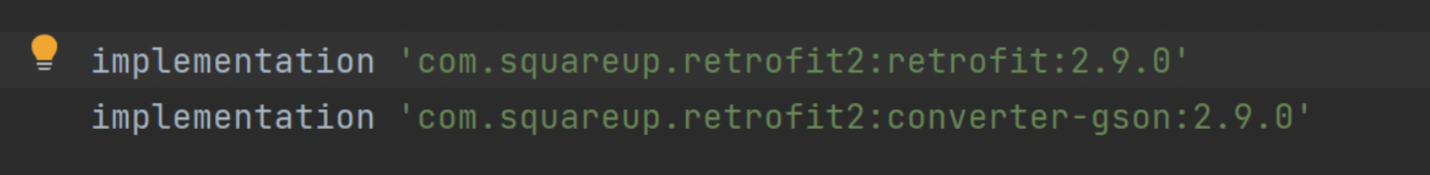
Description automatically generated with low confidence

adding Scroll Bar to the textview

A picture containing text, plaque, screenshot

Description automatically generated

Dependencies added to the project in the build.gradle file



User Entity used for casting the output response of the api

Graphical user interface, text, application

Description automatically generated

On successful rest-api response, converting the response to users list and displays o/p

Text

Description automatically generated

On Failure of getting the response from the rest-api the this function executes

Graphical user interface, text

Description automatically generated

**ICP-11: Text to Speech**

In this ICP, we have created an application which converts a text into a speech.

1. First we have added textview, edit text and a button to design the interface.

2. textview displays the tile of the icp. Edit text is a field which takes the input from the user which the application later converts it to a speech on clicking button named as speak.

3. The converstion process is taken care by text to speech api.

4. In this text to speech api, speak method converts the text to speech.

5. If the input entered by the user is unpronounceable or contains any error, then the error message “TextToSpeech cannot be translated” will be displayed.

**OUTPUT:**

A picture containing text, monitor, screenshot

Description automatically generated

Fetching the data using Id’s

Text

Description automatically generated

Text to speech logic which handles error and success scenario’s

Text

Description automatically generated