

```
1 package com.example.instagram
2
3 import android.annotation.SuppressLint
4 import android.graphics.ColorMatrix
5 import android.icu.text.CaseMap
6 import android.media.Image
7 import android.os.Bundle
8 import android.widget.Toast
9 import androidx.activity.ComponentActivity
10 import androidx.activity.compose.setContent
11 import androidx.compose.foundation.BorderStroke
12 import androidx.compose.foundation.Image
13 import androidx.compose.foundation.background
14 import androidx.compose.foundation.border
15 import androidx.compose.foundation.interaction.
    MutableInteractionSource
16 import androidx.compose.foundation.interaction.
    collectIsPressedAsState
17 import androidx.compose.foundation.layout.Arrangement
18 import androidx.compose.foundation.layout.Column
19 //import androidx.compose.foundation.layout.
    ColumnScopeInstance.align
20 import androidx.compose.foundation.layout.Row
21 import androidx.compose.foundation.layout.Spacer
22 import androidx.compose.foundation.layout.fillMaxSize
23 import androidx.compose.foundation.layout.
    fillMaxWidth
24 import androidx.compose.foundation.layout.height
25 import androidx.compose.foundation.layout.padding
26 import androidx.compose.foundation.layout.size
27 import androidx.compose.foundation.layout.width
28 import androidx.compose.foundation.
    rememberScrollState
29 import androidx.compose.foundation.shape.
    RoundedCornerShape
30 import androidx.compose.foundation.verticalScroll
31 import androidx.compose.material3.AlertDialog
32 import androidx.compose.material3.Button
33 import androidx.compose.material3.ButtonDefaults
34 import androidx.compose.material3.Card
35 import androidx.compose.material3.Divider
```

```
36 import androidx.compose.material3.  
    ExperimentalMaterial3Api  
37 import androidx.compose.material3.MaterialTheme  
38 import androidx.compose.material3.OutlinedTextField  
39 import androidx.compose.material3.Surface  
40 import androidx.compose.material3.Text  
41 import androidx.compose.material3.TextField  
42 import androidx.compose.material3.TextFieldColors  
43 import androidx.compose.material3.TextFieldDefaults  
44 import androidx.compose.runtime.Composable  
45 import androidx.compose.runtime.getValue  
46 import androidx.compose.runtime.mutableStateOf  
47 import androidx.compose.runtime.remember  
48 import androidx.compose.runtime.setValue  
49 import androidx.compose.ui.Alignment  
50 import androidx.compose.ui.Alignment.Companion.  
    CenterVertically  
51 import androidx.compose.ui.Modifier  
52 import androidx.compose.ui.graphics.Color  
53 import androidx.compose.ui.graphics.Color.Companion.  
    Green  
54 import androidx.compose.ui.graphics.Color.Companion.  
    LightGray  
55 import androidx.compose.ui.graphics.RectangleShape  
56 import androidx.compose.ui.graphics.Shape  
57 import androidx.compose.ui.graphics.painter.  
    BrushPainter  
58 import androidx.compose.ui.platform.LocalContext  
59 import androidx.compose.ui.res.painterResource  
60 import androidx.compose.ui.text.font.FontFamily  
61 import androidx.compose.ui.text.font.FontStyle  
62 import androidx.compose.ui.text.font.FontWeight  
63 import androidx.compose.ui.text.style.TextAlign  
64 //import androidx.compose.ui.text.style.  
    TextForegroundColor.Unspecified.brush  
65 //import androidx.compose.ui.tooling.data.EmptyGroup.  
    name  
66 import androidx.compose.ui.tooling.preview.Preview  
67 import androidx.compose.ui.unit.dp  
68 import androidx.compose.ui.unit.sp  
69 import com.example.instagram.ui.theme.InstagramTheme
```

```
70 import java.lang.Double.parseDouble
71 import java.lang.NumberFormatException
72 import java.nio.file.WatchEvent
73
74 class MainActivity : ComponentActivity() {
75     override fun onCreate(savedInstanceState: Bundle
76         ?) {
77         super.onCreate(savedInstanceState)
78         setContent {
79             InstagramTheme {
80                 // A surface container using the '
81                 background' color from the theme
82                 Surface {
83                     instaSignUp()
84                 }
85             }
86         }
87
88         @SuppressLint("RememberReturnType")
89         @OptIn(ExperimentalMaterial3Api::class)
90         @Composable
91         fun instaSignUp() {
92             var mob by remember {
93                 mutableStateOf("")
94             }
95             var fullName by remember {
96                 mutableStateOf("")
97             }
98             var userName by remember {
99                 mutableStateOf("")
100             }
101             var pass by remember {
102                 mutableStateOf("")
103             }
104             var signUpStatus by remember {
105                 mutableStateOf(false)
106             }
107             var showSuccess by remember {
108                 mutableStateOf(false)
```

```
109     }
110     val context= LocalContext.current
111
112     Card(modifier = Modifier
113         .background(Color.White)
114         .verticalScroll(rememberScrollState())) {
115         Column(horizontalAlignment = Alignment.
116             CenterHorizontally,
117             modifier = Modifier
118                 .background(Color.White)
119                 .padding(50.dp)
120         ) {
121             Image(painter = painterResource(id = R.
122                 drawable.instagram),
123                 contentDescription = "InstaIcon",
124                 modifier= Modifier
125                     .height(100.dp)
126                     .padding(
127                         top = 10.dp,
128                         bottom = 20.dp,
129                         start = 10.dp,
130                         end = 10.dp
131                     ))
132             Text(text = "Sign up to see photos and
133                 videos\nfrom your friends.",
134                 fontSize = 17.sp, color = Color.Gray
135                 , textAlign = TextAlign.Center,
136                 modifier = Modifier
137                     .fillMaxWidth()
138                     .padding(
139                         bottom = 15.dp,
140                         start = 5.dp,
141                         end = 5.dp
142                     ),
143                 lineHeight = 22.sp,
144                 fontWeight = FontWeight.W500,
145             )
146             Button(onClick = { /*TODO*/ },
147                 colors = ButtonDefaults.
148                 buttonColors(Color(0xFF0394F4)),
```

```

145                modifier = Modifier
146                    .fillMaxWidth()
147                    .padding(bottom = 15.dp),
148                shape = RoundedCornerShape(25)
149            ) {
150                Row() {
151                    Image(painter = painterResource(
152                        id = R.drawable.facebook),
153                        contentDescription = "
154                        facebookIcon",
155                        modifier = Modifier.size(21.
156                        dp)
157                    )
158                    Text(text = "    Log in with
159                    facebook",
160                        fontSize = 17.sp)
161                }
162            }
163
164            Row(modifier = Modifier
165                .fillMaxWidth()
166                .padding(bottom = 20.dp)) {
167                Spacer(modifier = Modifier
168                    .width(126.dp)
169                    .height(1.dp)
170                    .background(color = Color.Black)
171                    .align(alignment =
172                        CenterVertically))
173                Text(text = "    OR    ",
174                    fontSize = 15.sp,
175                    fontWeight = FontWeight.W600,
176                    color = Color.Gray)
177                Spacer(modifier = Modifier
178                    .width(126.dp)
179                    .height(1.dp)
180                    .background(color = Color.Black)
181                    .align(alignment =
182                        CenterVertically))
183            }
184
185            OutlinedTextField(value =mob ,

```

```

180                onChange = {mob = it},
181                modifier = Modifier
182                    .padding(bottom = 1.dp)
183                    .fillMaxWidth()
184                    .height(60.dp),
185                colors = TextFieldDefaults.
outlinedTextFieldColors(containerColor = Color(
    0x36E1E1E1),
186                focusedBorderColor = LightGray),
187                label = { Text(text = "Mobile Number
or Email",
188                    fontSize = 15.sp,
189                    color = Color(0x94000000)
190                ) }
191            )
192
193            OutlinedTextField(value = fullName,
194                onChange = {fullName = it},
195                modifier = Modifier
196                    .padding(bottom = 1.dp)
197                    .fillMaxWidth()
198                    .height(60.dp),
199                colors = TextFieldDefaults.
outlinedTextFieldColors(containerColor = Color(
    0x36E1E1E1),
200                focusedBorderColor = LightGray),
201                label = { Text(text = "Full Name",
202                    fontSize = 15.sp,
203                    color = Color(0x94000000)) }
204            )
205            OutlinedTextField(value = userName,
206                onChange = {userName = it},
207                modifier = Modifier
208                    .padding(bottom = 1.dp)
209                    .fillMaxWidth()
210                    .height(60.dp),
211                colors = TextFieldDefaults.
outlinedTextFieldColors(containerColor = Color(
    0x36E1E1E1),
212                focusedBorderColor = LightGray),
213                label = { Text(text = "Username",

```

```

214                fontSize=15.sp,
215                color = Color(0x94000000))})
216
217                OutlinedTextField(value =pass ,
218                onChange = {pass=it},
219                modifier = Modifier
220                .padding(bottom = 23.dp)
221                .fillMaxWidth()
222                .height(60.dp),
223                colors = TextFieldDefaults.
outlinedTextFieldColors(containerColor = Color(
0x36E1E1E1),
224                focusedBorderColor= LightGray),
225                label={ Text(text = "Password",
226                fontSize=15.sp,
227                color = Color(0x94000000))})
228
229                if(signUpStatus){
230                if(mobValid(mob)&&userName.
isEmpty()&&fullName.isEmpty()&&pass.length>=8
){
231
232                AlertDialog(onDismissRequest = {
signUpStatus=false},
233                title = { Text(text = "
Confirm sign up??")},
234                confirmButton={ Button(
onClick = {
235                signUpStatus=false
236                Toast.makeText(context,"
Signed up Successfully",Toast.LENGTH_LONG).show()
237                }) {
238                Text(text = "Yes")
239                })}
240                }
241                else if(!mobValid(mob)){
242                AlertDialog(onDismissRequest = {
signUpStatus=false },
243                title = { Text(text = "
Invalid Mobile Number or email")},
244                text={ Text(text = "Check

```

```

244 and Try again"}},
245             confirmButton = { Button(
                onClick = { signUpStatus=false}) {
246                 Text(text = "OK")
247             }
248         })
249     }
250     else if(userName.isEmpty()){
251         AlertDialog(onDismissRequest = {
            signUpStatus=false },
252             title = { Text(text = "
Invalid Full Name")},
253             text={ Text(text = "Check
and Try again")},
254             confirmButton = { Button(
                onClick = { signUpStatus=false}) {
255                 Text(text = "OK")
256             }
257         })
258     }
259     else if(fullName.isEmpty()){
260         AlertDialog(onDismissRequest = {
            signUpStatus=false },
261             title = { Text(text = "
Invalid User Name")},
262             text={ Text(text = "Check
and Try again")},
263             confirmButton = { Button(
                onClick = { signUpStatus=false}) {
264                 Text(text = "OK")
265             }
266         })
267     }
268     else if(pass.length<8){
269         AlertDialog(onDismissRequest = {
            signUpStatus=false },
270             title = { Text(text = "
Invalid Password")},
271             text={ Text(text = "Check
and Try again")},
272             confirmButton = { Button(

```



```

272 onClick = { signUpStatus=false}) {
273             Text(text = "OK")
274             }
275         })
276     }
277 }
278
279         Text(text = "People who use our service
may have uploaded your contact information to
Instagram. Learn More",
280             fontSize = 12.sp,
281             lineHeight = 16.sp,
282             textAlign = TextAlign.Center,
283             modifier = Modifier.padding(start=5.
dp,
284                                     end=5.dp, bottom = 18.dp)
285             )
286
287         Text(text = "By signing up, you agree to
our Terms , Privacy Policy and Cookies Policy.",
288             fontSize = 12.sp,
289             lineHeight = 16.sp,
290             textAlign = TextAlign.Center,
291             modifier = Modifier
292                 .padding(
293                     start = 5.dp,
294                     bottom = 18.dp, end = 5.dp
295                 )
296             .fillMaxWidth()
297         )
298
299         Button(onClick = { signUpStatus=true},
300             colors = ButtonDefaults.buttonColors
(Color(0xFF03A8F4)),
301             shape = RoundedCornerShape(30),
302             modifier = Modifier.fillMaxWidth()
303         ) {
304             Text("Sign up", fontSize = 17.sp)
305         }
306     }
307 }

```

```
308 }
309
310 @Preview(showBackground = true)
311 @Composable
312 fun GreetingPreview() {
313     InstagramTheme {
314         instaSignUp()
315     }
316 }
317
318 fun mobValid(mob:String):Boolean{
319     var numeric =true
320
321     try {
322         var num=parseDouble(mob)
323     }catch (e:NumberFormatException){
324         numeric=false
325     }
326
327     if(numeric&&mob.length!=10)numeric=false
328     else if(!numeric&&("@" in mob)&&(".com" in mob
329 )){
329         numeric=true
330     }
331     return numeric
332 }
```