File - C:\Users\saiteja\Android projects\keys(kotlin)\app\src\main\java\com\bst\keyskotlin\MainActivity.kt

```kotlin
package com.bst.keyskotlin

import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Email
import androidx.compose.material.icons.filled.Person
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedButton
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TextButton
import androidx.compose.material3.TextField
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
```

```
28 import androidx.compose.ui.Modifier
29 import androidx.compose.ui.layout.layoutId
30 import androidx.compose.ui.platform.LocalContext
31 import androidx.compose.ui.res.painterResource
32 import androidx.compose.ui.tooling.preview.Preview
33 import androidx.compose.ui.unit.dp
34 import androidx.constraintlayout.compose.ConstraintLayout
35 import androidx.constraintlayout.compose.ConstraintSet
36 import com.example.compose.AppTheme
37
38
39 class MainActivity : ComponentActivity() {
40     override fun onCreate(savedInstanceState: Bundle?) {
41         super.onCreate(savedInstanceState)
42         setContent {
43             AppTheme {
44                 // A surface container using the 'background' color from the theme
45                 Surface(
46                     modifier = Modifier.fillMaxSize(), color = MaterialTheme.
   colorScheme.background
47                 ) {
48                     contraintLayout()
49                 }
50             }
51         }
52     }
53
```

```
54      @OptIn(ExperimentalMaterial3Api::class)
55      @Composable
56      fun contraintLayout() {
57          var constraints = ConstraintSet {
58              var appLogo = createRefFor(id = "appLogo")
59              var loginBtn = createRefFor("loginBtn")
60              var signinBtn = createRefFor("signinBtn")
61              var forgotpasswordBtn = createRefFor("forgotpasswordBtn")
62              var nameTextView = createRefFor("nameTextView")
63              var emailTextView = createRefFor("emailTextView")
64              var nameTextField = createRefFor("nameTextField")
65              var emailTextField = createRefFor("emailTextField")
66
67              constrain(loginBtn) {
68                  top.linkTo(nameTextField.bottom)
69                  bottom.linkTo(parent.bottom)
70                  start.linkTo(parent.start)
71                  end.linkTo(parent.end)
72              }
73              constrain(signinBtn) {
74                  top.linkTo(loginBtn.top)
75                  bottom.linkTo(parent.bottom)
76                  start.linkTo(parent.start)
77                  end.linkTo(parent.end)
78              }
79
80              constrain(forgotpasswordBtn) {
```

```
 81                  top.linkTo(signinBtn.bottom)
 82                  bottom.linkTo(parent.bottom)
 83                  start.linkTo(parent.start)
 84                  end.linkTo(parent.end)
 85              }
 86
 87              constrain(nameTextView) {
 88                  top.linkTo(nameTextField.top)
 89                  end.linkTo(nameTextField.start)
 90                  bottom.linkTo(nameTextField.bottom)
 91
 92              }
 93
 94              constrain(emailTextView) {
 95                  top.linkTo(emailTextField.top)
 96                  end.linkTo(emailTextField.start)
 97                  bottom.linkTo(emailTextField.bottom)
 98
 99              }
100
101              constrain(nameTextField) {
102                  top.linkTo(parent.top)
103                  end.linkTo(parent.end)
104                  bottom.linkTo(parent.bottom)
105              }
106
107              constrain(emailTextField) {
```

```
                top.linkTo(nameTextField.top)
                bottom.linkTo(loginBtn.top)
                end.linkTo(parent.end)
            }

        }
        ConstraintLayout(constraints, modifier = Modifier.padding(all = 5.dp)) {
            val context = LocalContext.current
            var name by remember {
                mutableStateOf("")
            }
            var password by remember {
                mutableStateOf("")
            }
            OutlinedButton(
                onClick = { /*TODO*/ },
                modifier = Modifier
                    .layoutId("signinBtn")
                    .padding(all = 10.dp)
            ) {
                Text(text = "Sign in")

            }
            TextButton(
                onClick = { /*TODO*/ },
                modifier = Modifier
                    .layoutId("forgotpasswordBtn")
```

```
135                         .padding(all = 10.dp)
136                 ) {
137                     Text(text = "Forgot Password")
138                 }
139                 Button(
140                     onClick = {
141                         Toast.makeText(context, name, Toast.LENGTH_SHORT).show()
142                         name = ""
143                         val intent = Intent(context, MainActivity2::class.java)
144                         intent.putExtra("name", name)
145                         startActivity(intent)
146                     }, modifier = Modifier
147                         .layoutId("loginBtn")
148                         .padding(top = 20.dp)
149                 ) {
150                     Text(text = " Log in")
151                 }
152                 Text(
153                     text = "Name", modifier = Modifier
154                         .layoutId("nameTextView")
155                         .padding(all = 5.dp)
156                 )
157                 Text(
158                     text = "Email", modifier = Modifier
159                         .layoutId("emailTextView")
160                         .padding(all = 5.dp)
161                 )
```

```
162             TextField(value = name, onValueChange = { name = it }, leadingIcon = {
163                 Icon(imageVector = Icons.Default.Person, contentDescription = "
    person")
164             }, label = {
165                 Text(text = "Enter Username")
166             }, modifier = Modifier
167                 .layoutId("nameTextField")
168                 .padding(all = 20.dp)
169             )
170
171             TextField(value = password, onValueChange = { password = it },
    leadingIcon = {
172                 Icon(imageVector = Icons.Default.Email, contentDescription = "email
    ")
173             }, label = {
174                 Text(text = "Enter Email")
175             }, modifier = Modifier
176                 .layoutId("emailTextField")
177                 .padding(all = 20.dp)
178             )
179
180             Image(
181                 painter = painterResource(id = R.drawable.blood_donation),
182                 contentDescription = "appIcon",
183             )
184
185         }
```

```
186     }
187
188     @Preview
189     @Composable
190     fun SimpleComposablePreview() {
191         AppTheme {
192             // A surface container using the 'background' color from the theme
193             Surface(
194                 modifier = Modifier.fillMaxSize(), color = MaterialTheme.
    colorScheme.background
195             ) {
196                 contraintLayout()
197             }
198         }
199     }
200
201 }
202
203
204
```

File - C:\Users\saiteja\Android projects\keys(kotlin)\app\src\main\java\com\bst\keyskotlin\MainActivity2.kt

```kotlin
package com.bst.keyskotlin

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.selection.toggleable
import androidx.compose.material3.Button
import androidx.compose.material3.Checkbox
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.InputChip
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
```

File - C:\Users\saiteja\Android projects\keys(kotlin)\app\src\main\java\com\bst\keyskotlin\MainActivity2.kt

```
28 import androidx.compose.ui.layout.layoutId
29 import androidx.compose.ui.res.painterResource
30 import androidx.compose.ui.semantics.Role
31 import androidx.compose.ui.text.style.TextAlign
32 import androidx.compose.ui.tooling.preview.Preview
33 import androidx.compose.ui.unit.dp
34 import androidx.constraintlayout.compose.ConstraintLayout
35 import androidx.constraintlayout.compose.ConstraintSet
36 import androidx.constraintlayout.compose.Dimension
37 import com.example.compose.AppTheme
38
39 class MainActivity2 : ComponentActivity() {
40     override fun onCreate(savedInstanceState: Bundle?) {
41         super.onCreate(savedInstanceState)
42         setContent {
43             AppTheme() {
44                 // A surface container using the 'background' color from the theme
45                 Surface(
46                     modifier = Modifier.fillMaxSize(),
47                     color = MaterialTheme.colorScheme.background
48                 ) {
49                     GetIntentData()
50                 }
51             }
52         }
53     }
54
```

```
55      @OptIn(ExperimentalMaterial3Api::class)
56      @Composable
57      fun GetIntentData() {
58          val constraints = ConstraintSet {
59              val imageWelcome = createRefFor("imageWelcome")
60              val buttonSignIn = createRefFor("buttonSubmit")
61              val textHeadLine = createRefFor("textHeadLine")
62              val assistChipA = createRefFor("assistChipA")
63              val assistChipB = createRefFor("assistChipB")
64              val assistChipO = createRefFor("assistChipO")
65              val assistChipAB = createRefFor("assistChipAB")
66              val assistChipPlus = createRefFor("assistChipPlus")
67              val assistChipMinus = createRefFor("assistChipMinus")
68              val checkBoxLayout = createRefFor("checkBoxLayout")
69              constrain(imageWelcome) {
70                  top.linkTo(parent.top)
71                  start.linkTo(parent.start)
72                  end.linkTo(parent.end)
73                  height = Dimension.value(150.dp)
74                  width = Dimension.value(150.dp)
75              }
76              constrain(textHeadLine) {
77                  top.linkTo(imageWelcome.bottom)
78                  start.linkTo(parent.start)
79                  end.linkTo(parent.end)
80              }
81
```

```
 82                constrain(assistChipA) {
 83                    top.linkTo(textHeadLine.bottom)
 84                    start.linkTo(parent.start)
 85                    width = Dimension.percent(0.5F)
 86                    height = Dimension.value(100.dp)
 87                }
 88                constrain(assistChipB) {
 89                    top.linkTo(assistChipA.top)
 90                    end.linkTo(parent.end)
 91                    start.linkTo(assistChipA.end)
 92                    width = Dimension.percent(0.5F)
 93                    height = Dimension.value(100.dp)
 94                }
 95                constrain(assistChipO) {
 96                    top.linkTo(assistChipA.bottom)
 97                    start.linkTo(parent.start)
 98                    width = Dimension.percent(0.5F)
 99                    height = Dimension.value(100.dp)
100                }
101                constrain(assistChipAB) {
102                    top.linkTo(assistChipO.top)
103                    end.linkTo(parent.end)
104                    start.linkTo(assistChipO.end)
105                    width = Dimension.percent(0.5F)
106                    height = Dimension.value(100.dp)
107                }
108                constrain(assistChipPlus) {
```

```
109                 top.linkTo(assistChipO.bottom)
110                 end.linkTo(assistChipO.end)
111             }
112             constrain(assistChipMinus) {
113                 top.linkTo(assistChipAB.bottom)
114                 start.linkTo(assistChipAB.start)
115             }
116             constrain(checkBoxLayout) {
117                 bottom.linkTo(buttonSignIn.top)
118             }
119             constrain(buttonSignIn) {
120                 bottom.linkTo(parent.bottom)
121                 start.linkTo(parent.start)
122                 end.linkTo(parent.end)
123                 end.linkTo(parent.end)
124                 width = Dimension.matchParent
125             }
126         }
127         ConstraintLayout(
128             constraints,
129             modifier = Modifier
130                 .fillMaxSize()
131                 .padding(all = 5.dp)
132         ) {
133             Image(
134                 painter = painterResource(id = R.drawable.imge_blood_drop),
135                 contentDescription = "Welcome image",
```

```
136                     modifier = Modifier
137                         .layoutId("imageWelcome")
138                         .padding(top = 50.dp)
139                 )
140                 Text(
141                     text = "Please pick your blood type",
142                     modifier = Modifier
143                         .layoutId("textHeadLine"),
144                     style = MaterialTheme.typography.headlineLarge,
145                     textAlign = TextAlign.Center
146                 )
147 
148                 var chipASelectedValue by remember {
149                     mutableStateOf(false)
150                 }
151                 var chipBSelectedValue by remember {
152                     mutableStateOf(false)
153                 }
154                 var chipOSelectedValue by remember {
155                     mutableStateOf(false)
156                 }
157                 var chipABSelectedValue by remember {
158                     mutableStateOf(false)
159                 }
160                 InputChip(
161                     selected = chipASelectedValue,
162                     onClick = {
```

```
163                         chipASelectedValue = !chipASelectedValue
164                         if (chipBSelectedValue) {
165                             chipBSelectedValue = !chipBSelectedValue
166                         } else if (chipOSelectedValue) {
167                             chipOSelectedValue = !chipOSelectedValue
168                         } else if (chipABSelectedValue) {
169                             chipABSelectedValue = !chipABSelectedValue
170                         }
171                     },
172                     label = {
173                         Text(
174                             text = "A", textAlign = TextAlign.Center,
175                             modifier = Modifier.fillMaxWidth()
176                         )
177                     },
178                     modifier = Modifier
179                         .layoutId("assistChipA")
180                         .padding(all = 10.dp)
181                 )
182 
183                 InputChip(
184                     selected = chipBSelectedValue,
185                     onClick = {
186                         chipBSelectedValue = !chipBSelectedValue
187                         if (chipASelectedValue) {
188                             chipASelectedValue = !chipASelectedValue
189                         } else if (chipOSelectedValue) {
```

```
                            chipOSelectedValue = !chipOSelectedValue
                        } else if (chipABSelectedValue) {
                            chipABSelectedValue = !chipABSelectedValue
                        }
                    },
                    label = {
                        Text(
                            text = "B", textAlign = TextAlign.Center,
                            modifier = Modifier.fillMaxWidth()
                        )
                    },
                    modifier = Modifier
                        .layoutId("assistChipB")
                        .padding(all = 10.dp)
                )

                InputChip(
                    selected = chipOSelectedValue,
                    onClick = {
                        chipOSelectedValue = !chipOSelectedValue
                        if (chipBSelectedValue) {
                            chipBSelectedValue = !chipBSelectedValue
                        } else if (chipASelectedValue) {
                            chipASelectedValue = !chipASelectedValue
                        } else if (chipABSelectedValue) {
                            chipABSelectedValue = !chipABSelectedValue
                        }
```

```
217                     },
218                     label = {
219                         Text(
220                             text = "O", textAlign = TextAlign.Center,
221                             modifier = Modifier.fillMaxWidth()
222                         )
223                     },
224                     modifier = Modifier
225                         .layoutId("assistChipO")
226                         .padding(all = 10.dp)
227                 )
228 
229                 InputChip(
230                     selected = chipABSelectedValue,
231                     onClick = {
232                         chipABSelectedValue = !chipABSelectedValue
233                         if (chipBSelectedValue) {
234                             chipBSelectedValue = !chipBSelectedValue
235                         } else if (chipOSelectedValue) {
236                             chipOSelectedValue = !chipOSelectedValue
237                         } else if (chipASelectedValue) {
238                             chipASelectedValue = !chipASelectedValue
239                         }
240                     },
241                     label = {
242                         Text(
243                             text = "AB", textAlign = TextAlign.Center,
```

```
244                             modifier = Modifier.fillMaxWidth()
245                         )
246                     },
247                     modifier = Modifier
248                         .layoutId("assistChipAB")
249                         .padding(all = 10.dp)
250                 )
251                 var chipPlusSelectedValue by remember {
252                     mutableStateOf(false)
253                 }
254                 var chipMinusSelectedValue by remember {
255                     mutableStateOf(false)
256                 }
257                 InputChip(
258                     selected = chipPlusSelectedValue,
259                     onClick = {
260                         chipPlusSelectedValue = !chipPlusSelectedValue
261                         if (chipMinusSelectedValue) chipMinusSelectedValue = !
    chipMinusSelectedValue
262                     },
263                     label = {},
264                     leadingIcon = {
265                         Icon(
266                             painter = painterResource(id = R.drawable.baseline_plus_24
    ),
267                             contentDescription = "Plus",
268                         )
```

```
269                 },
270                 modifier = Modifier
271                     .layoutId("assistChipPlus")
272                     .padding(all = 10.dp)
273             )
274
275
276             InputChip(
277                 selected = chipMinusSelectedValue,
278                 onClick = {
279                     chipMinusSelectedValue = !chipMinusSelectedValue
280                     if (chipPlusSelectedValue) chipPlusSelectedValue = !
    chipPlusSelectedValue
281                 },
282                 label = {},
283                 leadingIcon = {
284                     Icon(
285                         painter = painterResource(id = R.drawable.
    baseline_horizontal_rule_24),
286                         contentDescription = "Plus",
287                     )
288                 },
289                 modifier = Modifier
290                     .layoutId("assistChipMinus")
291                     .padding(all = 10.dp)
292             )
293             val (checkedState, onStateChange) = remember { mutableStateOf(true) }
```

```
294                 Row(
295                     Modifier
296                         .fillMaxWidth()
297                         .height(56.dp)
298                         .layoutId("checkBoxLayout")
299                         .toggleable(
300                             value = checkedState,
301                             onValueChange = { onStateChange(!checkedState) },
302                             role = Role.Checkbox
303                         )
304                         .padding(horizontal = 16.dp),
305                     verticalAlignment = Alignment.CenterVertically
306                 ) {
307                     Checkbox(
308                         checked = checkedState,
309                         onCheckedChange = null // null recommended for accessibility
    with screenreaders
310                     )
311                     Text(
312                         text = "I want to receive the notification about the blood
    donation camp",
313                         style = MaterialTheme.typography.bodyLarge,
314                         modifier = Modifier.padding(start = 16.dp)
315                     )
316                 }
317                 Button(
318                     onClick = {
```

```
319
320                 },
321                 modifier = Modifier
322                     .layoutId("buttonSubmit")
323                     .padding(start = 60.dp, end = 60.dp, top = 10.dp, bottom = 50.
    dp)
324             ) {
325                 Text(text = "Submit")
326             }
327         }
328     }
329
330     @Preview
331     @Composable
332     fun preview() {
333         GetIntentData()
334     }
335 }
336
337
```