**Kanishka Ghosh**

**20BPS1125**

**Week 2 Assignment**

**MainActivity.kt**

package com.example.bloodbankapp\_20bps1125  
  
import android.content.Intent  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.Image  
import androidx.compose.foundation.background  
import androidx.compose.foundation.layout.Arrangement  
import androidx.compose.foundation.layout.Box  
import androidx.compose.foundation.layout.Column  
import androidx.compose.foundation.layout.Spacer  
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.layout.width  
import androidx.compose.foundation.shape.RoundedCornerShape  
import androidx.compose.material3.Button  
import androidx.compose.material3.ButtonDefaults  
import androidx.compose.material3.Card  
import androidx.compose.material3.CardDefaults  
import androidx.compose.material3.ExperimentalMaterial3Api  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.material3.TextField  
import androidx.compose.material3.TextFieldDefaults  
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  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.layout.ContentScale  
import androidx.compose.ui.platform.*LocalContext*import androidx.compose.ui.res.painterResource  
import androidx.compose.ui.text.input.PasswordVisualTransformation  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import com.example.bloodbankapp\_20bps1125.ui.theme.BloodBankApp\_20BPS1125Theme  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** BloodBankApp\_20BPS1125Theme **{** // A surface container using the 'background' color from the theme  
 Surface(  
 modifier = Modifier.*fillMaxSize*(),  
 color = MaterialTheme.colorScheme.background  
 ) **{** LoginPage()  
 **}  
 }  
 }** }  
}  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun LoginPage() {  
 Box(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*background*(*Color*(252, 169, 169, 255)),  
// contentAlignment = Alignment.Center  
 ) **{** val context = *LocalContext*.current  
 var username by remember **{** *mutableStateOf*("") **}** var password by remember **{** *mutableStateOf*("") **}** var showError by remember **{** *mutableStateOf*(false) **}** Column(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*padding*(horizontal = 16.*dp*),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Image(  
 painter = painterResource(id = R.drawable.*logo*),  
 contentDescription = null,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*height*(200.*dp*),  
 contentScale = ContentScale.Crop  
 )  
 Spacer(modifier = Modifier.*height*(36.*dp*))  
 Card(  
 elevation = CardDefaults.cardElevation(  
 defaultElevation = 10.*dp* ),  
 shape = *RoundedCornerShape*(40.*dp*),  
 ) **{** Column(  
 verticalArrangement = Arrangement.Center,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*height*(400.*dp*)  
 .*background*(*Color*(253, 75, 75, 255))  
 .*padding*(all = 10.*dp*)  
 ) **{** TextField(  
 value = username,  
 onValueChange = **{** username = **it }**,  
 label = **{** Text("Username") **}**,  
 modifier = Modifier.*fillMaxWidth*(),  
 shape = *RoundedCornerShape*(15.*dp*),  
 colors = TextFieldDefaults.outlinedTextFieldColors(  
 focusedBorderColor = Color.Transparent,  
 focusedLabelColor = Color.Transparent,  
 placeholderColor = Color.Transparent,  
 containerColor = *Color*(248, 166, 166, 255),  
 ),  
 )  
 Spacer(modifier = Modifier.*height*(30.*dp*))  
 TextField(  
 value = password,  
 onValueChange = **{** password = **it }**,  
 label = **{** Text("Password") **}**,  
 visualTransformation = PasswordVisualTransformation(),  
 modifier = Modifier.*fillMaxWidth*(),  
 shape = *RoundedCornerShape*(15.*dp*),  
 colors = TextFieldDefaults.outlinedTextFieldColors(  
 focusedBorderColor = Color.Transparent,  
 focusedLabelColor = Color.Transparent,  
 placeholderColor = Color.Transparent,  
 containerColor = *Color*(248, 166, 166, 255),  
 ),  
 )  
 if (showError) {  
 Text(  
 text = "Invalid username or password",  
 color = Color.Red,  
 modifier = Modifier.*padding*(top = 8.*dp*)  
 )  
 }  
 Spacer(modifier = Modifier.*height*(30.*dp*))  
 Text(  
 text = "Forgot Password?",  
 style = MaterialTheme.typography.bodySmall,  
 color = Color.White  
 )  
 Spacer(modifier = Modifier.*height*(50.*dp*))  
 Column(  
 modifier=Modifier.*fillMaxWidth*(),  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Button(  
 onClick = **{** if (username == "admin" && password == "admin") {  
 showError = false  
 context.startActivity(Intent(context, HomeActivity::class.*java*))  
 } else {  
 showError = true  
 }  
 **}**,  
 colors = ButtonDefaults.buttonColors(*Color*(50, 30, 30)),  
 modifier = Modifier  
 .*width*(150.*dp*)  
 .*height*(70.*dp*)  
 ) **{** Text(text = "Login")  
 **}  
 }  
 }  
 }  
 }  
 }**}  
  
@Preview(showBackground = true)  
@Composable  
fun LoginPreview() {  
 BloodBankApp\_20BPS1125Theme **{** LoginPage()  
 **}**}

**HomeActivity.kt**

package com.example.bloodbankapp\_20bps1125  
  
import android.annotation.SuppressLint  
import android.app.DatePickerDialog  
import android.content.Intent  
import android.os.Build  
import android.os.Bundle  
import android.widget.DatePicker  
import android.widget.Toast  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.annotation.RequiresApi  
import androidx.compose.foundation.BorderStroke  
import androidx.compose.foundation.border  
import androidx.compose.foundation.layout.Arrangement  
import androidx.compose.foundation.layout.Column  
import androidx.compose.foundation.layout.ExperimentalLayoutApi  
import androidx.compose.foundation.layout.PaddingValues  
import androidx.compose.foundation.layout.Row  
import androidx.compose.foundation.layout.Spacer  
import androidx.compose.foundation.layout.consumedWindowInsets  
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.layout.size  
import androidx.compose.foundation.layout.width  
import androidx.compose.foundation.rememberScrollState  
import androidx.compose.foundation.shape.RoundedCornerShape  
import androidx.compose.foundation.verticalScroll  
import androidx.compose.material.icons.Icons  
import androidx.compose.material.icons.filled.*Favorite*import androidx.compose.material.icons.filled.*Home*import androidx.compose.material.icons.filled.*Menu*import androidx.compose.material.icons.filled.*Person*import androidx.compose.material.icons.filled.*Phone*import androidx.compose.material.icons.filled.*Send*import androidx.compose.material3.BottomAppBar  
import androidx.compose.material3.Button  
import androidx.compose.material3.ButtonDefaults  
import androidx.compose.material3.DrawerValue  
import androidx.compose.material3.ExperimentalMaterial3Api  
import androidx.compose.material3.Icon  
import androidx.compose.material3.IconButton  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.ModalDrawerSheet  
import androidx.compose.material3.ModalNavigationDrawer  
import androidx.compose.material3.NavigationDrawerItem  
import androidx.compose.material3.NavigationDrawerItemDefaults  
import androidx.compose.material3.Scaffold  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.material3.TextField  
import androidx.compose.material3.TextFieldDefaults  
import androidx.compose.material3.TopAppBar  
import androidx.compose.material3.rememberDrawerState  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.MutableState  
import androidx.compose.runtime.mutableStateOf  
import androidx.compose.runtime.remember  
import androidx.compose.runtime.rememberCoroutineScope  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.draw.drawBehind  
import androidx.compose.ui.geometry.Offset  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.graphics.vector.ImageVector  
import androidx.compose.ui.platform.*LocalContext*import androidx.compose.ui.text.font.FontFamily  
import androidx.compose.ui.text.font.FontWeight  
import androidx.compose.ui.text.input.PasswordVisualTransformation  
import androidx.compose.ui.text.input.TextFieldValue  
import androidx.compose.ui.text.input.VisualTransformation  
import androidx.compose.ui.text.style.TextAlign  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
import com.example.bloodbankapp\_20bps1125.ui.theme.BloodBankApp\_20BPS1125Theme  
import kotlinx.coroutines.launch  
import java.util.Calendar  
import java.util.Date  
  
class HomeActivity : ComponentActivity() {  
 @RequiresApi(Build.VERSION\_CODES.*O*)  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** BloodBankApp\_20BPS1125Theme **{** Surface(  
 modifier = Modifier.*fillMaxSize*(),  
 color = MaterialTheme.colorScheme.background  
 ) **{** App()  
 **}  
 }  
 }** }  
}  
  
sealed class Screen(val route: String, val title: String, val icon: ImageVector) {  
 object Home : Screen("home", "Home", Icons.Default.*Home*)  
 object ContactUs : Screen("contact\_us", "Contact Us", Icons.Default.*Phone*)  
 object BookAppointment : Screen("book\_appointment", "Book Appointment", Icons.Default.*Send*)  
}  
  
@OptIn(ExperimentalMaterial3Api::class)  
@RequiresApi(Build.VERSION\_CODES.*O*)  
@Composable  
fun App() {  
 val screens = *listOf*(Screen.Home, Screen.ContactUs, Screen.BookAppointment)  
 val drawerState = rememberDrawerState(DrawerValue.*Closed*)  
 val scope = rememberCoroutineScope()  
 val selectedItem = remember **{** *mutableStateOf*(screens[0]) **}** return(  
 ModalNavigationDrawer(  
 drawerState = drawerState,  
 drawerContent = **{** ModalDrawerSheet **{** Spacer(Modifier.*height*(12.*dp*))  
 screens.*forEach* **{** item **->** NavigationDrawerItem(  
 icon = **{** Icon(item.icon, contentDescription = null) **}**,  
 label = **{** Text(item.title) **}**,  
 selected = item == selectedItem.value,  
 onClick = **{** scope.*launch* **{** drawerState.close() **}** selectedItem.value = item  
 **}**,  
 modifier = Modifier.*padding*(NavigationDrawerItemDefaults.ItemPadding)  
 )  
 **}  
 }  
 }**,  
 content = **{** Scaffold(  
 topBar = **{** TopAppBar(  
 title = **{** val context = *LocalContext*.current  
 Row(  
 verticalAlignment = Alignment.CenterVertically,  
 horizontalArrangement = Arrangement.SpaceBetween,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 ) **{** IconButton(onClick = **{** scope.*launch* **{** drawerState.open() **} }**) **{** Icon(  
 Icons.Filled.*Menu*,  
 contentDescription = "",  
 modifier = Modifier  
 .*size*(30.*dp*)  
 )  
 **}** Spacer(modifier = Modifier)  
 Text(text = "Donate Blood")  
 Button(  
 onClick = **{** context.startActivity(  
 Intent(  
 context,  
 MainActivity::class.*java* )  
 )  
 **}**, colors = ButtonDefaults.buttonColors(  
 containerColor = Color.White,  
 contentColor = Color.Red  
 )  
 ) **{** Text(text = "Logout")  
 **}  
 }  
 }**,  
 modifier = Modifier.*drawBehind* **{** drawLine(  
 Color.Red,  
 *Offset*(0f, size.height),  
 *Offset*(size.width, size.height),  
 5f  
 )  
 **}** )  
 **}**,  
 bottomBar = **{** CustomBottomBar() **}**,  
 content = **{** pad **->** MainContent(pad) **}**,  
 )  
 **}** )  
 )  
}  
  
@OptIn(ExperimentalMaterial3Api::class)  
@RequiresApi(Build.VERSION\_CODES.*O*)  
@Composable  
fun CustomScaffold() {  
 Scaffold(topBar = **{** CustomTopBar() **}**,  
 bottomBar = **{** CustomBottomBar() **}**,  
 content = **{** pad **->** MainContent(pad) **}**,  
 )  
}  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun CustomTopBar() {  
 TopAppBar(  
 title = **{** val context = *LocalContext*.current  
 Row(  
 verticalAlignment = Alignment.CenterVertically,  
 horizontalArrangement = Arrangement.SpaceBetween,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 ) **{** Icon(  
 Icons.Filled.*Menu*, contentDescription = "", modifier = Modifier  
 .*size*(30.*dp*)  
 )  
 Spacer(modifier = Modifier)  
 Text(text = "Donate Blood")  
 Button(onClick = **{** context.startActivity(Intent(context, MainActivity::class.*java*))  
 **}**,colors = ButtonDefaults.buttonColors(  
 containerColor = Color.White,  
 contentColor = Color.Red)) **{** Text(text = "Logout")  
 **}  
 }  
 }**,  
 modifier = Modifier.*drawBehind* **{** drawLine(  
 Color.Red,  
 *Offset*(0f, size.height),  
 *Offset*(size.width, size.height),  
 5f  
 )  
 **}** )  
}  
  
@Composable  
fun CustomBottomBar() {  
 remember **{** *mutableStateOf*(0) **}** BottomAppBar(  
 modifier = Modifier.*drawBehind* **{** drawLine(  
 Color.White,  
 *Offset*(0f, 0f),  
 *Offset*(size.width, 0f),  
 4f  
 )  
 **}**,  
 containerColor = *Color*(126, 23, 23),  
 ) **{** Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.SpaceEvenly,  
 verticalAlignment =  
 Alignment.CenterVertically  
 ) **{** Icon(imageVector = Icons.Default.*Home*, "", modifier = Modifier.*size*(30.*dp*),tint = Color.White)  
 Icon(imageVector = Icons.Default.*Favorite*, "", modifier = Modifier.*size*(30.*dp*),tint = Color.White)  
 Icon(imageVector = Icons.Default.*Person*, "", modifier = Modifier.*size*(30.*dp*),tint = Color.White)  
 **}  
 }**}  
  
@SuppressLint("RememberReturnType")  
@RequiresApi(Build.VERSION\_CODES.*O*)  
@OptIn(ExperimentalLayoutApi::class)  
@Composable  
fun MainContent(padding: PaddingValues) {  
  
 val mContext = *LocalContext*.current  
  
 // Declaring integer values  
 // for year, month and day  
 val mYear: Int  
 val mMonth: Int  
 val mDay: Int  
  
 // Initializing a Calendar  
 val mCalendar = Calendar.getInstance()  
  
 // Fetching current year, month and day  
 mYear = mCalendar.get(Calendar.*YEAR*)  
 mMonth = mCalendar.get(Calendar.*MONTH*)  
 mDay = mCalendar.get(Calendar.*DAY\_OF\_MONTH*)  
  
 mCalendar.*time* = Date()  
  
 // Declaring a string value to  
 // store date in string format  
 val mDate = remember **{** *mutableStateOf*("") **}** val primaryTextColor = remember **{** *mutableStateOf*(*Color*(115, 115, 115))  
 **}** val tertiaryTextColor = remember **{** *mutableStateOf*(Color.Black)  
 **}** val textFieldColor = remember **{** *mutableStateOf*(*Color*(250, 250, 250))  
 **}** val mobile = remember **{** *mutableStateOf*(TextFieldValue())  
 **}** val fullName = remember **{** *mutableStateOf*(TextFieldValue())  
 **}** val mDatePickerDialog = DatePickerDialog(  
 mContext,  
 **{** \_: DatePicker, mYear: Int, mMonth: Int, mDayOfMonth: Int **->** mDate.value = "$mDayOfMonth/${mMonth+1}/$mYear"  
 **}**, mYear, mMonth, mDay  
 )  
  
 val email = remember **{** *mutableStateOf*(TextFieldValue())  
 **}** val address = remember **{** *mutableStateOf*(TextFieldValue())  
 **}** val bloodGroup = remember **{** *mutableStateOf*(TextFieldValue())  
 **}** Column(  
 modifier = Modifier.*verticalScroll*(rememberScrollState())  
 ) **{** Column(  
 modifier = Modifier  
 .*padding*(20.*dp*)  
 .*padding*(padding)  
 .*consumedWindowInsets*(padding),  
 horizontalAlignment = Alignment.CenterHorizontally  
 ) **{** Spacer(modifier = Modifier.*height*(20.*dp*))  
 Text(  
 "Be the reason for someone’s heartbeat.",  
 color = *Color*(206, 24, 24, 255),  
 fontFamily =  
 FontFamily.SansSerif,  
 fontSize = 20.*sp*,  
 fontWeight = FontWeight.SemiBold,  
 textAlign = TextAlign.Center  
 )  
  
 Spacer(modifier = Modifier.*height*(20.*dp*))  
 CustomTextField(  
 modifier = Modifier.*fillMaxWidth*(),  
 mutableValue =  
 fullName,  
 label = "Full Name",  
 focusedColor = primaryTextColor.value,  
 textColor = tertiaryTextColor.value,  
 conColor = textFieldColor.value  
  
 )  
 Spacer(modifier = Modifier.*height*(10.*dp*))  
  
 CustomTextField(  
 modifier = Modifier.*fillMaxWidth*(),  
 mutableValue =  
 email,  
 label = "Email",  
 focusedColor = primaryTextColor.value,  
 textColor = tertiaryTextColor.value,  
 conColor = textFieldColor.value  
 )  
 Spacer(modifier = Modifier.*height*(10.*dp*))  
  
 CustomTextField(  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*height*(100.*dp*),  
 mutableValue =  
 address,  
 label = "Address",  
 focusedColor = primaryTextColor.value,  
 textColor = tertiaryTextColor.value,  
 conColor = textFieldColor.value  
 )  
 Spacer(modifier = Modifier.*height*(10.*dp*))  
  
 CustomTextField(  
 modifier = Modifier.*fillMaxWidth*(),  
 mutableValue =  
 mobile,  
 label = "Mobile Number",  
 focusedColor = primaryTextColor.value,  
 textColor = tertiaryTextColor.value,  
 conColor = textFieldColor.value  
 )  
  
 Spacer(modifier = Modifier.*height*(10.*dp*))  
 CustomTextField(  
 modifier = Modifier.*fillMaxWidth*(),  
 mutableValue =  
 bloodGroup,  
 label = "Blood Group",  
 focusedColor = primaryTextColor.value,  
 textColor = tertiaryTextColor.value,  
 conColor = textFieldColor.value  
 )  
 Spacer(modifier = Modifier.*height*(10.*dp*))  
 Button(onClick = **{** mDatePickerDialog.show()  
 **}**,  
 colors = ButtonDefaults.buttonColors(  
 containerColor = *Color*(182, 35, 35),  
 contentColor = Color.White  
 )) **{** Text(text = "Choose Appointment Date", color = Color.White)  
 **}** Spacer(modifier = Modifier.*height*(10.*dp*))  
 val context= *LocalContext*.current  
 CustomButton(  
 buttonText = "Book Appointment", isLogo = true, onClick = **{** if(fullName.toString() !="")  
 Toast.makeText(context,"Appointment Booked",Toast.*LENGTH\_SHORT*).show()  
 else  
 Toast.makeText(context,"Enter all the details",Toast.*LENGTH\_SHORT*).show()  
 **}** )  
 **}  
 }**}  
  
@OptIn(ExperimentalMaterial3Api::class)  
@Composable  
fun CustomTextField(  
 modifier: Modifier = Modifier,  
 mutableValue: MutableState<TextFieldValue>, label: String,  
 placeholder: String  
 = label,  
 focusedColor: Color, conColor: Color = *Color*(250, 250, 250),  
 isHideVal: Boolean = false, textColor: Color  
) {  
 TextField(  
 modifier = modifier.*border*(  
 *BorderStroke*(0.2.*dp*, focusedColor), *RoundedCornerShape* (4.*dp*)  
 ),  
 value = mutableValue.value,  
 onValueChange = **{** mutableValue.value = **it }**,  
 label = **{** Text(text = label) **}**,  
 placeholder = **{** Text(text = placeholder) **}**,  
 colors = TextFieldDefaults.outlinedTextFieldColors(  
 focusedBorderColor = Color.Transparent,  
 focusedLabelColor = focusedColor,  
 placeholderColor = focusedColor,  
 textColor = textColor,  
 unfocusedBorderColor = Color.Transparent,  
 unfocusedLabelColor = focusedColor,  
 unfocusedLeadingIconColor = focusedColor,  
 focusedLeadingIconColor = focusedColor,  
 containerColor = conColor,  
  
 ),  
 visualTransformation = if (isHideVal) PasswordVisualTransformation(  
 mask =  
 '\u2022'  
 ) else VisualTransformation.None,  
  
 )  
}  
  
@Composable  
fun CustomButton(  
 buttonText: String,  
 textColor: Color = Color.White,  
 backgroundColor: Color = *Color*(0, 149, 246),  
 onClick: () -> Unit = **{}**,  
 isLogo: Boolean = false  
) {  
 Button(  
 onClick = onClick,  
 shape = *RoundedCornerShape*(10.*dp*),  
 colors = ButtonDefaults.buttonColors(*Color*(182, 35, 35, 255)),  
 modifier = Modifier.*fillMaxWidth*()  
 ) **{** if (isLogo) {  
 Icon(  
 Icons.Filled.*Send*, contentDescription =  
 "facebook logo", modifier = Modifier.*size*(25.*dp*)  
 )  
 }  
 Spacer(modifier = Modifier.*width*(10.*dp*))  
 Text(  
 buttonText,  
 color = textColor,  
 fontSize = 16.*sp*,  
 fontWeight = FontWeight.Bold  
 )  
 **}**}  
  
@RequiresApi(Build.VERSION\_CODES.*O*)  
@Preview(showBackground = true)  
@Composable  
fun GreetingPreview() {  
 BloodBankApp\_20BPS1125Theme **{** App()  
 **}**}





