class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

val context = LocalContext.current

var task by remember {

mutableStateOf("")

}

var taskList by remember {

mutableStateOf(mutableListOf<String>())

}

var affich = remember {

mutableStateOf(false)

}

Column(

horizontalAlignment =Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center

) {

Spacer(modifier = Modifier.height(20.dp))

Text(text = "ADD A TASK", fontWeight = FontWeight.Bold, fontFamily = FontFamily.Serif, fontSize = 25.sp)

Row(

horizontalArrangement =Arrangement.Center,

verticalAlignment = Alignment.CenterVertically

) {

Spacer(modifier = Modifier.width(10.dp))

OutlinedTextField(value =task , onValueChange = {newTask -> task = newTask}, placeholder = { Text(

text = "Enter the task")},

colors = TextFieldDefaults.outlinedTextFieldColors(

focusedBorderColor = Color.Blue,

unfocusedBorderColor = Color.Black

)

)

Spacer(modifier = Modifier.width(5.dp))

Button(onClick = {

if(task.isNotBlank()){

taskList +=task

task = ""

}else{

Toast.makeText(context,"Error please enter the task !",Toast.LENGTH\_SHORT).show()

}

},

colors = ButtonDefaults.buttonColors(backgroundColor = Color.Blue,

contentColor = Color.White

)

) {

Text(text = "ADD")

}

}

Spacer(modifier = Modifier.height(30.dp))

Column(

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center

) {

Text(text = "LIST OF PROGRAMMED TASK", fontSize = 20.sp, fontFamily = FontFamily.Serif, fontWeight = FontWeight.Bold)

Divider(modifier = Modifier

.width(width = 300.dp)

.height(height = 5.dp)

,

color = Color.Blue

)

Spacer(modifier = Modifier

.width(30.dp)

)

Spacer(modifier = Modifier.height(30.dp))

viewTask(tasks = taskList)

}

}

}

}

}

@Composable

fun viewLigne(num:Int,description:String){

Row(

verticalAlignment = Alignment.CenterVertically,

horizontalArrangement = Arrangement.Center

) {

Text(text = num.toString(),

color = Color.White,

modifier = Modifier

.padding(10.dp)

.drawBehind {

drawCircle(

color = Color.Blue,

radius = this.size.maxDimension

)

}

)

Spacer(modifier = Modifier.width(10.dp))

Text(text = description, fontSize = 20.sp)

}

}

@Composable

fun viewTask(tasks:List<String>){

var n = 1

LazyColumn{

items(tasks){task ->

viewLigne(num = n, description = task)

n++

}

}

}