

Source code: Kotlin

```
/**
 * this is the implementation of the Intro screen
 * the UI was implemented in activity_intro.xml file
 */
class IntroActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_intro)
        window.decorView.systemUiVisibility = View.SYSTEM_UI_FLAG_FULLSCREEN
        var atg = AnimationUtils.loadAnimation(this, R.anim.atg)
        iv_picstatistic.startAnimation(atg)
        tv_welcome.startAnimation(AnimationUtils.loadAnimation(this, R.anim.bgone))
        tv_description.startAnimation(AnimationUtils.loadAnimation(this, R.anim.bgone))
        l_buttons.startAnimation(AnimationUtils.loadAnimation(this, R.anim.bgtwo))

        var currentUserID = FirestoreClass().getCurrentUserID()

        //STAY LOGGED IN
        //intent to TractivityMain activity as soon as the user enter the app if the user is logged in
        already
        if(currentUserID.isNotEmpty()){
            startActivity(Intent(this, TractivityMain::class.java))
            finish()
        }

        //direct user to the Login Screen when login button is clicked
        bt_login.setOnClickListener{
            val intent = Intent(this, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }

        //direct user to the Sign-Up Screen when sign up button is clicked
        bt_signUp.setOnClickListener{
            val intent = Intent(this, SignUpActivity::class.java)
            startActivity(intent)
            finish()
        }
    }
}
```

*Login Screen of the application

* GUI designed in activity_login.xml

*/

```
class LoginActivity : BaseActivity() {
```

```
    private lateinit var auth: FirebaseAuth
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.activity_login)
```

```
        window.decorView.systemUiVisibility = View.SYSTEM_UI_FLAG_FULLSCREEN
```

```
        auth = FirebaseAuth
```

```
        bt_login.setOnClickListener{
```

```
            loginUser()
```

```
        }
```

```
        tv_signUP.setOnClickListener{
```

```
            val intent = Intent(this, SignUpActivity::class.java)
```

```
            startActivity(intent)
```

```
            finish()
```

```
        }
```

```
        tv_forgot_pass.setOnClickListener{
```

```
            startActivity(Intent(this, PasswordResetActivity::class.java))
```

```
        }
```

```
    }
```

```
//logs in the existing user
```

```

private fun loginUser (){
    val email:String = et_email_login.text.toString().trim{ it <= ' ' }
    val password:String = et_password_login.text.toString()
    if(validateLogin(email,password)){
        auth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(this) { task ->
                if (task.isSuccessful) {
                    Log.d("Login", "LoginWithEmail:success")
                    val user = auth.currentUser
                    Toast.makeText(this,
                        "Welcome back to Tractivity",Toast.LENGTH_SHORT).show()
                    val intent = Intent(this, TractivityMain::class.java)
                    startActivity(intent)
                    finish()

                } else {
                    // If sign in fails, display a error message to the user.
                    Log.w("Login", "LoginWithEmail:failure", task.exception)
                    Toast.makeText(baseContext, "Authentication failed, incorrect email or password was
typed",
                        Toast.LENGTH_SHORT).show()
                }
            }
    }
}

//validates the login form
private fun validateLogin ( email: String ,
    password: String) :Boolean{
    when {

```

```

        TextUtils.isEmpty(email) ->{
            showError("Please enter your email address")
            return false
        }
        TextUtils.isEmpty(password) ->{
            showError("Please enter your password")
            return false
        }
        else ->{
            return true
        }
    }
}
}

/**
 * The sign Screen of the application was implemented here
 * with Gui designed in activity_sign_up.xml file
 */
class SignUpActivity : BaseActivity() {

    private lateinit var auth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_sign_up)
        window.decorView.systemUiVisibility = View.SYSTEM_UI_FLAG_FULLSCREEN
        auth = FirebaseAuth

        bt_SignUp.setOnClickListener {

```

```

        createUser()
    }

    //Intend to login page
    tv_login.setOnClickListener {
        val intent = Intent(this, LoginActivity::class.java)
        startActivity(intent)
        finish()
    }
}

// user is created and stored on authentication server and firestore
private fun createUser(){
    val name: String = et_name.text.toString().trim{ it <= ' ' }
    val email: String = et_email.text.toString().trim{ it <= ' ' }
    val password : String = et_password.text.toString()
    val reTypedPassword : String = et_reTypePassword.text.toString()

    if(validateSignUp(name,email,password,reTypedPassword)){
        auth.createUserWithEmailAndPassword(email,password)
            .addOnCompleteListener(this){
                task ->
                if (task.isSuccessful) {
                    val user = UserClass(Firebase.auth.currentUser!!.uid,name,email)

                    FireStoreClass().registerUserOnDB(user)

                    Toast.makeText(this,
                        "Welcome $name to Tractivity",Toast.LENGTH_SHORT).show()

                    val intent = Intent(this, TractivityMain::class.java)
                    startActivity(intent)
                }
            }
    }
}

```

```

        finish()
    }else {
        Toast.makeText(this,task.exception!!.message.toString(),Toast.LENGTH_LONG).show()
    }
}
}
}
}

```

//function to validate the sighUp form

```

private fun validateSignUp (name:String , email: String ,
                            password: String, reTypedPassword : String) :Boolean{
    when {
        TextUtils.isEmpty(name) ->{
            showError("Please enter your name")
            return false
        }
        TextUtils.isEmpty(email) ->{
            showError("Please enter your email address")
            return false
        }
        TextUtils.isEmpty(password) ->{
            showError("Please enter a password")
            return false
        }
        TextUtils.isEmpty(reTypedPassword) ->{
            showError("Please re-Type your password")
            return false
        }
        !isPasswordMatched(password,reTypedPassword) ->{
            return false
        }
    }
}

```

```

        else ->{
            return true
        }
    }
}

//checks if both the passwords matches
private fun isPasswordMatched (password:String, reTypedPassword:String) :Boolean{
    return if(password == reTypedPassword){
        true
    }else{
        showError("Passwords do not match, please retype")
        false
    }
}
}

/**
 * this class contains the of the functionality the home screen, Tractivity
 * The stop watch, app navigation and storing the activity data in the database was implemented
 here.
 * the ui was designed in the activity_trmain.xml, activity_stopwatch.xml, dialog_save.xml
 */
@Suppress("DEPRECATION")
class TractivityMain : BaseActivity(), NavigationView.OnNavigationItemSelectedListener {

    private var pauseTime : Long = 0
    private var running : Boolean = false
    private var activitySubmitted : Boolean = false
    val firestore = Firebase.firestore

```

```
var progress : String ?= null
var activityName : String = ""
var isActivityStarted : Boolean = false
var isActivitySaved : Boolean = false
var isActivitySelected: Boolean = false
```

```
val CHANNEL_ID = "channel_id"
val CHANNEL_NAME = "channel_name"
val NOTIFICATION_ID = 1
```

```
@SuppressWarnings("SetTextI18n")
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_trmain)
```

```
//HIDE STATUS BAR
hideStatusBar()
```

```
//CUSTOM ACTION BAR
setupActionBar()
```

```
//NAVIGATION LISTENER
navigation.setNavigationItemSelectedListener(this)
```

```
val notificationIntent = Intent(this, TractivityMain::class.java)
notificationIntent.setAction(Intent.ACTION_MAIN)

val pendingIntent : PendingIntent =
PendingIntent.getActivity(this,0,notificationIntent,PendingIntent.FLAG_UPDATE_CURRENT)
```



```

createNotificationChannel()

// to build the notification panel
val buildNotification = NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.ic_stat_name)
    .setContentTitle("Tractivity")
    .setContentText("is running")
    .setNotificationSilent()
    .setContentIntent(pendingIntent)
    .setPriority(NotificationCompat.PRIORITY_DEFAULT).build()

// start and pause the activity
btn_start.setOnClickListener {
    isActivityStarted = true
    NotificationManagerCompat.from(this).notify(NOTIFICATION_ID,buildNotification)
    startAndPauseActivity()
}

//STOP STOPWATCH and call Save Activity Dialog
btn_stop.setOnClickListener{
    if( isActivityStarted){
        NotificationManagerCompat.from(this).cancel(NOTIFICATION_ID)
        progress = c_chronometer.text.toString()
        pauseStopWatch()
        saveDialogFunction()

    }else{
        Toast.makeText(this, "Activity not started", Toast.LENGTH_SHORT).show()
    }
}

```

```

    }

    // Reset stopwatch and confirmation dialog
    btn_reset.setOnClickListener {
        if(isActivityStarted){
            pauseStopWatch()
            confirmationDialog()
        }

    }

    populateNavUsername()

}

// start and pause the activity function
private fun startAndPauseActivity(){
    if (!running) {
        startStopWatch()
    } else {
        pauseStopWatch()
    }

}

}

// START THE STOPWATCH
private fun startStopWatch(){
    iArrow.startAnimation(AnimationUtils.loadAnimation(this, R.anim.rotating_arrow))
    c_chronometer.base = SystemClock.elapsedRealtime() + pauseTime

```

```
c_chronometer.start()
running = true
btn_start.text = "Pause"
}
```

```
// PAUSE THE STOPWATCH
```

```
private fun pauseStopWatch(){
    iArrow.clearAnimation()
    pauseTime = c_chronometer.base - SystemClock.elapsedRealtime()
    c_chronometer.stop()
    running = false
    btn_start.text= "Resume"
}
```

```
//reset the stopwatch
```

```
private fun resetStopwatch(){

    c_chronometer.base= SystemClock.elapsedRealtime()
    iArrow.clearAnimation()
    pauseTime = 0
    running=false
    btn_start.text="Start"
}
```

```
//display a confirmation dialog
```

```
private fun confirmationDialog(){
    val confBuilder = AlertDialog.Builder(this)
    confBuilder.setTitle("Reset stopwatch?")
    confBuilder.setMessage("resetting the stopwatch will cancel the progress")
    confBuilder.setPositiveButton("RESET") { dialog: DialogInterface, i: Int ->
        resetStopwatch()
    }
}
```

```

        isActivityStarted=false
        dialog.cancel()
    }
    confBuilder.setNegativeButton("Cancel") { dialog: DialogInterface, i: Int ->
        dialog.cancel()
    }
    confBuilder.show()
}

//populate username on navigation
fun populateNavUsername(){
    FirebaseFirestore.getInstance().collection(Constants.USERS)
        .document(FireStoreClass().getCurrentUserID()).get()
        .addOnSuccessListener {documentSnapshot ->
            val user = documentSnapshot.toObject<UserClass>()!!
            val navHeader = navigation.getHeaderView(0)
            val navUsername = navHeader.findViewById<TextView>(R.id.tv_username)
            navUsername.text = user.name

        }
    }

//prevent accidental exit
override fun onBackPressed() {
    doubleBackExit()
}

//CUSTOM ACTION BAR
private fun setupActionBar() {
    setSupportActionBar(toolbar_trmain)

```

```

toolbar_trmain.setNavigationIcon(R.drawable.ic_menu)

toolbar_trmain.setNavigationOnClickListener {
    toggleDrawer()
}
}

```

```
//TOGGLE NAVIGATION DRAWER
```

```

private fun toggleDrawer() {
    if (drawer_layout.isDrawerOpen(GravityCompat.START)) {
        drawer_layout.closeDrawer(GravityCompat.START)
    } else {
        drawer_layout.openDrawer(GravityCompat.START)
    }
}

```

// Intend to the selected activity screen from the navigation menu. Menus were defined in the menu/drawer_menu.xml file

```

override fun onNavigationItemSelected(item: MenuItem): Boolean {
    when (item.itemId){
        R.id.main_pg -> {
            drawer_layout.closeDrawer(GravityCompat.START)
            return true
        }
        R.id.activities ->{
            val intent = Intent(this, ActivitiesActivity::class.java)
            startActivity(intent)
        }
        R.id.charts ->{
            val intent = Intent(this, ChartsActivity::class.java)
            startActivity(intent)
        }
    }
}

```

```

R.id.profile ->{
    val intent = Intent(this, UserProfileActivity::class.java)
    startActivity(intent)
}

R.id.logout ->{
    FirebaseAuth.getInstance().signOut()
    val intent = Intent(this, IntroActivity::class.java)
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP or Intent.FLAG_ACTIVITY_NEW_TASK)
    startActivity(intent)
    finish()
}
}

drawer_layout.closeDrawer(GravityCompat.START)
return true
}

```

```

//creating a notification channel
private fun createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {

        val importance = NotificationManager.IMPORTANCE_DEFAULT
        val channel1 = NotificationChannel(CHANNEL_ID, CHANNEL_NAME, importance).apply {
            //behaviour of the notification
            setSound(null,null)
        }

        // Register the channel with the system
        val notificationManager: NotificationManager =
            getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
        notificationManager.createNotificationChannel(channel1)
    }
}

```

```
}
```

```
//SAVE ACTIVITY CUSTOM DIALOG inflated from dialog_save.xml file
```

```
@SuppressWarnings("SetTextI18n")
```

```
private fun saveDialogFunction() {
```

```
    val saveDialogView = LayoutInflater.from(this).inflate(R.layout.dialog_save,null)
```

```
    saveDialogView.tv_description.text= "Following $progress is spent on:"
```

```
    val saveDialogbuilder = AlertDialog.Builder(this)
```

```
        .setView(saveDialogView)
```

```
    val saveActivityDialog = saveDialogbuilder.show()
```

```
    saveActivityDialog.setCancelable(false) // prevent user to close the dialog by clicking outside the dialog
```

```
    saveDialogView.bt_selectActivity.setOnClickListener {
```

```
        // ACTIVITY NAMES DIALOG populated with the activity lists retrieved from the DB
```

```
        firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
```

```
            .collection(Constants.ACTIVITIES)
```

```
            .get()
```

```
            .addOnSuccessListener { activities ->
```

```
                Log.d("DB", "All documents received")
```

```
                val activityNameList = ArrayList<String>()
```

```
                for (activity in activities) {
```

```
                    val activityClass: ActivityClass? = activity.toObject(ActivityClass::class.java)
```

```
                    if (activityClass != null){
```

```
                        activityNameList.add(activityClass.name)
```

```
                }
```

```
            }
```

```
            val items:Array<String> = activityNameList.toArray()
```

```
            val selectActivityBuilder = AlertDialog.Builder(this)
```

```

        selectActivityBuilder.setTitle("Choose activity")
        selectActivityBuilder.setSingleChoiceItems(items,-1){
            dialogInterface: DialogInterface, i :Int ->

            saveDialogView.et_activityName.setText(items[i])
            isActivitySelected = true
            dialogInterface.dismiss()

        }
        selectActivityBuilder.setNeutralButton("cancel"){ dialog:DialogInterface, which->
            dialog.cancel()
        }
        val activitySelectDialog = selectActivityBuilder.create()
        activitySelectDialog.show()
    }
    .addOnFailureListener { exception ->
        Log.d("DB", "Error getting documents: ", exception)
    }

}

// storing the data on the firestore once the submit button in clicked
saveDialogView.bt_submit.setOnClickListener{
    activityName = saveDialogView.et_activityName.text.toString()
    if (TextUtils.isEmpty(activityName)){
        Toast.makeText(applicationContext, "Enter or select Activity Name",
        Toast.LENGTH_SHORT).show()
    } else {
        val activity = ActivityClass(activityName)
        val record = ActivityRecordClass(parseProgress(progress!!!))
        if(isActivitySelected){
            FireStoreClass().saveRecordOnDB(record,activity.name)
        }
    }
}

```



```

    } else{
        FirestoreClass().saveActivityOnDB(activity)
        FirestoreClass().saveRecordOnDB(record,activity.name)
    }
    isActivitySelected = false
    isActivitySelected = false
    resetStopwatch()
    saveActivityDialog.dismiss()
}

}

saveDialogView.bt_cancel.setOnClickListener{
    Toast.makeText(applicationContext, "Activity not saved", Toast.LENGTH_SHORT).show()
    saveActivityDialog.dismiss()
}

}

//PARSE STRING PROGRESS TO LONG (mm:ss or hh:mm:ss)
fun parseProgress (progress:String) : Long{
    var progressInSeconds :Long = 0
    val strArray : Array<String> = progress.split(":").toTypedArray()
    if(strArray.size<3){
        progressInSeconds += strArray[0].toLong()*60
        progressInSeconds += strArray[1].toLong()
        return progressInSeconds
    }
    progressInSeconds += strArray[0].toLong()*3600
    progressInSeconds += strArray[1].toLong()*60
    progressInSeconds += strArray[2].toLong()
}

```

```

        return progressInSeconds
    }

}

/**
 * The activity list screen was implemented in this file
 * the GUI was designed in the activity_activities.xml file
 * resources used: https://www.youtube.com/watch?v=6Gm3eMG8KqI
                   https://www.youtube.com/watch?v=afl\_i6uvvU0
                   https://guides.codepath.com/android/using-the-recyclerview
 */
@Suppress("DEPRECATION")
class ActivitiesActivity : AppCompatActivity() {
    private val firestore = Firebase.firestore

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_activities)
        window.setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN
        )
        displayActivityListFromDB()
        iv_back.setOnClickListener{
            onBackPressed()
        }

        iv_add.setOnClickListener{
            val intent = Intent(this, AddActivity::class.java)
            startActivity(intent)
        }
    }
}

```

```

        finish()
    }

}

// POPULATE RECYCLEVIEW
private fun populateActivityListRv (activityList: ArrayList<ActivityClass>){
    if (activityList.size> 0){
        rv_activitiesList.layoutManager = LinearLayoutManager(this)
        rv_activitiesList.setHasFixedSize(true)

        val adapter = ActivityAdapter(activityList)
        rv_activitiesList.adapter = adapter

        adapter.setOnItemClickListener(object: ActivityAdapter.OnItemClickListener {
            override fun onItemClick(position: Int, item: ActivityClass) {
                val intent = Intent(this@ActivitiesActivity, DetailsActivity::class.java)
                intent.putExtra(Constants.ACTIVITYNAME,item.name)
                startActivity(intent)
            }
        })
    }else{
        Toast.makeText(this, "no activity stored", Toast.LENGTH_SHORT).show()
    }
}

// retrieve the activity details from the firestore and display them
private fun displayActivityListFromDB(){
    firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
        .collection(Constants.ACTIVITIES)

```

```

        .get()
        .addOnSuccessListener { results ->
            Log.d("DB", "All documents received")
            val activityList = ArrayList<ActivityClass>()
            for (result in results) {
                val activity: ActivityClass? = result.toObject(ActivityClass::class.java)
                if (activity != null) {
                    activityList.add(activity)
                }
            }
            populateActivityListRv(activityList)
        }.addOnFailureListener { exception ->
            Log.d("DB", "Error getting documents: ", exception)
        }
    }
}

/**
 * Add/Created activity manually was implemnted in this class
 * the gui design: activity_add_activity.xml
 */
class AddActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_add_activity)

        window.setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN
        )

        setupActionBar()
    }
}

```

```

//store the activity on the firestore
bt_add.setOnClickListener{
    val activityName = et_activityName.text.toString()
    val description = et_description.text.toString()
    val dueDate = et_dueDate.text.toString()
    if (TextUtils.isEmpty(activityName)) {
        Toast.makeText(
            applicationContext,
            "Please Enter Activity Name",
            Toast.LENGTH_SHORT
        ).show()
    } else {
        val activity = ActivityClass(activityName,
            SimpleDateFormat("dd/MM/yyyy", Locale.getDefault()).format(Date())
            ,description, dueDate)
        FirestoreClass().saveActivityOnDB(activity)
        val intent = Intent(this, ActivitiesActivity::class.java)
        startActivity(intent)
        finish()
    }
}

}

//Custom app bar
private fun setupActionBar() {
    setSupportActionBar(toolbar_addActivity)
    val actionBar = supportActionBar
    actionBar!!.setDisplayHomeAsUpEnabled(true)
    actionBar.setHomeAsUpIndicator(R.drawable.ic_back)
    actionBar.title = "Add Activity"
}

```

```

        toolbar_addActivity.setNavigationOnClickListener{
            val intent = Intent(this, ActivitiesActivity::class.java)
            startActivity(intent)
            finish()

        }
    }
}

/**
 * some of the basic functionalities that were used in more than one class were implemented in this
 * class and inherited from.
 */
@Suppress("DEPRECATION")
open class BaseActivity : AppCompatActivity() {
    private var backPressedOnce = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_base)

    }

    fun hideStatusBar(){
        window.setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN
        )
    }

    fun getCurrentUserID():String{

```

```

        return FirebaseAuth.getInstance().currentUser!!.uid

    }

    //prevent user from accidentally exiting the app

    fun doubleBackExit() {
        if(backPressedOnce){
            super.onBackPressed()
            return
        }else{
            this.backPressedOnce = true
            Toast.makeText(this, "press back again to exit",Toast.LENGTH_SHORT).show()
            Handler().postDelayed({backPressedOnce = false},2000 )
        }
    }

    //display error message as snack bar

    fun showError(errorMessage:String){
        val snackBar=
        Snackbar.make(findViewById(android.R.id.content),errorMessage,Snackbar.LENGTH_LONG)
        snackBar.show()
    }

    /**
     * the activity details screen was implented in this class with the GUI designed in activity_details.xml
     file
     */
    class DetailsActivity : AppCompatActivity() {

```

```

private val firestore = Firebase.firestore
private lateinit var activityName : String

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_details)

    window.setFlags(
        WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN
    )
    if(intent.hasExtra(Constants.ACTIVITYNAME)){
        activityName = intent.getStringExtra(Constants.ACTIVITYNAME)!!
    }

    displayActivityDetailsFromDB()
    displayRecordListFromDB()
    displayProgress()

    iv_back.setOnClickListener{
        onBackPressed()
    }
    iv_delete.setOnClickListener{
        confirmationDialog()
    }
    iv_edit.setOnClickListener{
        displayActivityEditDialog()
    }
}

//query the progress records from the database and display in the recycleView

```



```

fun displayRecordListFromDB(){
    firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
        .collection(Constants.ACTIVITIES).document(activityName).collection(Constants.RECORDS)
        .get()
        .addOnSuccessListener { records ->
            Log.d("DB", "All documents received")
            val recordList = ArrayList<ActivityRecordClass>()
            for (record in records) {
                val record: ActivityRecordClass? = record.toObject(ActivityRecordClass::class.java)
                if (record != null) {
                    recordList.add(record)
                }
            }
            populateRecordsRv(recordList)
        }.addOnFailureListener { exception ->
            Log.d("DB", "Error getting documents: ", exception)
        }
}

```

```

//populate teh record RecyclerView with retrieved data from the firestore
private fun populateRecordsRv (recordList: ArrayList<ActivityRecordClass>) {
    if (recordList.size> 0){
        rv_recordList.layoutManager = LinearLayoutManager(this)
        rv_recordList.setHasFixedSize(true)

        val adapter = RecordAdapter(recordList)
        rv_recordList.adapter = adapter

    }
}

```

```

//query the details of the activity clicked from the database
private fun displayActivityDetailsFromDB (){
    firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
        .collection(Constants.ACTIVITIES).document(activityName).get()
        .addOnSuccessListener{ document ->
            Log.d("DB", "document received")
            val activity : ActivityClass? = document.toObject(ActivityClass::class.java)
            populateActivityDetails(activity!!)

        }.addOnFailureListener { exception ->
            Log.d("DB", "Error getting the document: ", exception)
        }
    }
}

```

```

// Populate the details sections
@SuppressLint("SetTextI18n")
fun populateActivityDetails(activity:ActivityClass){
    tv_activityName.text=activity.name
    tv_description.text= activity.description
    tv_startedDate.text= "Started Date: ${activity.date}"
    tv_dueDate.text= "Due date: ${activity.dueDate}"

}

```

```

// Delete an activity from the database
private fun deleteActivity(){
    firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
        .collection(Constants.ACTIVITIES).document(activityName).delete()
        .addOnSuccessListener {
            Log.d("DATA", "Activity has been deleted")
        }
    }
}

```

```

        Toast.makeText(this,
            "Activity has been deleted", Toast.LENGTH_SHORT).show()
    }
    .addOnFailureListener { e -> Log.w("DATA", "Error deleting document", e) }
}

//display and confirmation dialog before deleting the activity
private fun confirmationDialog(){
    val confBuilder = AlertDialog.Builder(this)
    confBuilder.setTitle("Delete Activity?")
    confBuilder.setPositiveButton("YES") { dialog: DialogInterface, i: Int ->
        deleteActivity()
        val intent = Intent(this, ActivitiesActivity::class.java)
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP or Intent.FLAG_ACTIVITY_NEW_TASK)
        startActivity(intent)
        finish()
        dialog.cancel()
    }
    confBuilder.setNegativeButton("NO") { dialog: DialogInterface, i: Int ->
        dialog.cancel()
    }
    confBuilder.show()
}

//Display a custom dialog to let user edit the activity details
// the dialog was designed in the dialog_edit_activity.xml
private fun displayActivityEditDialog(){
    val editDialogView = LayoutInflater.from(this).inflate(R.layout.dialog_edit_activity,null)
    firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
        .collection(Constants.ACTIVITIES).document(activityName).get()
        .addOnSuccessListener{ document ->

```

```

Log.d("DB", "document received")

val activity : ActivityClass? = document.toObject(ActivityClass::class.java)

if (activity != null) {
    editDialogView.et_description.setText(activity.description)
    editDialogView.et_dueDate.setText(activity.dueDate)
    val editDialogBuilder =AlertDialog.Builder(this).setView(editDialogView)
    val editDialog = editDialogBuilder.show()
    editDialogView.bt_submit.setOnClickListener {
        val description = editDialogView.et_description.text.toString()
        val dueDate = editDialogView.et_dueDate.text.toString()

        firestore.collection(Constants.USERS)
            .document(FireStoreClass().getCurrentUserID())
            .collection(Constants.ACTIVITIES).document(activityName).update(
                mapOf(
                    "description" to description,
                    "dueDate" to dueDate
                )
            )

        val intent = Intent(this, ActivitiesActivity::class.java)
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP or
Intent.FLAG_ACTIVITY_NEW_TASK)
        startActivity(intent)
        finish()

        Toast.makeText(this, "Activity details changed", Toast.LENGTH_SHORT).show()
        editDialog.dismiss()
    }

    editDialogView.bt_cancel.setOnClickListener {
        editDialog.cancel()
    }
}

}.addOnFailureListener { exception ->

```

```

        Log.d("DB", "Error getting the document: ", exception)
    }
}

//display the total time spent in a ctivity
@SuppressLint("SetTextI18n")
fun displayProgress(){
    firestore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())

.collection(Constants.ACTIVITIES).document(activityName).collection(Constants.RECORDS).get()
    .addOnSuccessListener{ documents ->
        Log.d("DB", "collection received")
        var totalTime = 0L
        for (document in documents){
            val record : ActivityRecordClass = document.toObject(ActivityRecordClass::class.java)
            totalTime += record.progress
        }
        tv_progress.text= "Total time spent: ${formatProgress(totalTime)}"
    }.addOnFailureListener { exception ->
        Log.d("DB", "Error getting the document: ", exception)
    }
}

```

```

//format the total progress from seconds to string in hh mm ss format
private fun formatProgress(progressInSec : Long) : String{
    val hours = progressInSec / 3600
    val minutes = (progressInSec % 3600) / 60
    val seconds = progressInSec % 60
    return when {
        hours >0 -> {
            String.format("%02dh %02dm %02ds", hours, minutes, seconds)
        }
    }
}

```

```

    }

    minutes>0 -> {
        String.format("%02dm %02ds",minutes, seconds)
    }

    else -> {
        String.format("%02ds", seconds)
    }
}

}

}

/**
 * Displaying PieCharts and bar charts were implemented here
 * with the GUI implemented in the activity.charts.xml
 */
class ChartsActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_charts)

        window.setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN
        )

        setupActionBar()
        showPieChart()
        showBarChart()
    }
}

```

```
//query and display activity data in bar chart
```

```
private fun showBarChart(){
```

```
FirestoreClass().fireStore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID()  
)
```

```
    .collection(Constants.ACTIVITIES)
```

```
    .get()
```

```
    .addOnSuccessListener { documents ->
```

```
        Log.d("DB", "All documents received")
```

```
        val activityBarEntries = ArrayList<BarEntry>()
```

```
        val activityBarLabels = ArrayList<String>()
```

```
        var index = 0f
```

```
        for (document in documents) {
```

```
            var totalProgress: Long = 0
```

```
            val activity = document.toObject(ActivityClass::class.java)
```

```
FirestoreClass().fireStore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID()  
)
```

```
    .collection(Constants.ACTIVITIES).document(activity.name).collection(
```

```
        Constants.RECORDS)
```

```
    .get().addOnSuccessListener { results ->
```

```
        for (result in results){
```

```
            val record : ActivityRecordClass = result.toObject(ActivityRecordClass::class.java)
```

```
            totalProgress += record.progress
```

```
        }
```

```
        if (totalProgress > 0){
```

```

        activityBarEntries.add(BarEntry(index++,totalProgress.toFloat()))

        activityBarLabels.add(activity.name)
    }

    createBarChart(activityBarEntries,activityBarLabels)
}.addOnFailureListener { exception ->
    Log.d("DB", "Error getting records: ", exception)
}

}

}.addOnFailureListener { exception ->
    Log.d("DB", "Error getting activities: ", exception)
}
}

//query and display activity data in Pie chart
private fun showPieChart(){

FirestoreClass().fireStore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID()
)

    .collection(Constants.ACTIVITIES)
    .get()
    .addOnSuccessListener { documents ->
        Log.d("DB", "All documents received")

        val activityPieData = ArrayList<PieEntry>()

        for (document in documents) {
            var totalProgress: Long = 0

            val activity = document.toObject(ActivityClass::class.java)

```



```
FirestoreClass().fireStore.collection(Constants.USERS).document(FireStoreClass().getCurrentUserID())
)
```

```
    .collection(Constants.ACTIVITIES).document(activity.name).collection(
        Constants.RECORDS)
    .get().addOnSuccessListener { results ->
        for (result in results){

            val record : ActivityRecordClass = result.toObject(ActivityRecordClass::class.java)
            totalProgress += record.progress
        }
    }
```

```
    if (totalProgress > 0){
        activityPieData.add(PieEntry(totalProgress.toFloat(), activity.name))
    }
```

```
    createPieChart(activityPieData)
}.addOnFailureListener { exception ->
    Log.d("DB", "Error getting records: ", exception)
}
```

```
}
```

```
}.addOnFailureListener { exception ->
    Log.d("DB", "Error getting activities: ", exception)
}
```

```
}
```

```
// Populate the Barchart view with defined features
```

```
private fun createBarChart(activityBarEntries: ArrayList<BarEntry>, activityBarLabels :
ArrayList<String>){
```

```
    var colors :ArrayList<Int> = ArrayList<Int>()
```

```

    for (color in ColorTemplate.JOYFUL_COLORS){
        colors.add(color)
    }
    for (color in ColorTemplate.COLORFUL_COLORS){
        colors.add(color)
    }
    val barDataSet = BarDataSet(activityBarEntries,"Bar data")
    barDataSet.value textSize=10f
    barDataSet.valueTextColor = Color.BLACK
    barDataSet.colors=colors
    barDataSet.valueFormatter= ProgressValueFormatter()

    val barData = BarData(barDataSet)
    crt_barChart.setDrawGridBackground(false)
    crt_barChart.setFitBars(true)
    crt_barChart.xAxis.valueFormatter= IndexAxisValueFormatter(activityBarLabels)
    crt_barChart.xAxis.position = XAxis.XAxisPosition.BOTTOM
    crt_barChart.description.isEnabled= false
    crt_barChart.data = barData

    crt_barChart.axisRight.isEnabled = false
    crt_barChart.invalidate()

}

// Populate the Pie Chart view with defined features
private fun createPieChart(activityData: ArrayList<PieEntry>){

    var colors :ArrayList<Int> = ArrayList<Int>()
    for (color in ColorTemplate.JOYFUL_COLORS){
        colors.add(color)
    }

```

```

    }
    for (color in ColorTemplate.COLORFUL_COLORS){
        colors.add(color)
    }

    val pieDataSet = PieDataSet(activityData,"Progress")
    pieDataSet.valueFormatter= ProgressValueFormatter()
    pieDataSet.colors = colors
    pieDataSet.valueTextSize = 10f
    pieDataSet.valueTextColor = Color.BLACK
    pieDataSet.sliceSpace= 2f
    pieDataSet.xValuePosition = PieDataSet.ValuePosition.OUTSIDE_SLICE


    val pieData = PieData(pieDataSet)
    crt_pieChart.data = pieData
    crt_pieChart.setEntryLabelColor(Color.BLACK)
    crt_pieChart.legend.isWordWrapEnabled = true
    crt_pieChart.animateXY(50, 50)
    crt_pieChart.description.isEnabled = false
    crt_pieChart.centerText="Time Spent"
    crt_pieChart.invalidate()

}

//CUSTOM ACTION BAR
private fun setupActionBar() {
    setSupportActionBar(toolbar_charts)
    val actionBar = supportActionBar
    actionBar!!.setDisplayHomeAsUpEnabled(true)
    actionBar.setHomeAsUpIndicator(R.drawable.ic_back)
    actionBar.title = "Charts"
}

```

```

        toolbar_charts.setNavigationOnClickListener{
            onBackPressed() }
    }

    //A class to override the chart value format
    private class ProgressValueFormatter : ValueFormatter(){
        override fun getFormattedValue(value:Float):String {
            return formatProgress(value.toLong())
        }

        private fun formatProgress(progressInSec : Long) : String{
            val hours = progressInSec / 3600
            val minutes = (progressInSec % 3600) / 60
            val seconds = progressInSec % 60
            return when {
                hours >0 -> {
                    String.format("%02dh%02dm%02ds", hours, minutes, seconds)
                }
                minutes>0 -> {
                    String.format("%02dm%02ds",minutes, seconds)
                }
                else -> {
                    String.format("%02ds", seconds)
                }
            }
        }
    }
}

import android.app.AlertDialog
import android.content.ContentValues
import android.os.Bundle

```

```

import android.util.Log
import android.view.LayoutInflater
import android.widget.Toast
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.firestore.ktx.firestore
import com.google.firebase.ktx.Firebase
import com.tanvir.tractivity.R
import com.tanvir.tractivity.constants.Constants
import com.tanvir.tractivity.model.UserClass
import kotlinx.android.synthetic.main.activity_user_profile.*
import kotlinx.android.synthetic.main.dialog_change_pass.*
import kotlinx.android.synthetic.main.dialog_change_pass.view.*

/**
 * this class contains teh implementation on the profile screen
 * the Gui was designed in teh activity_user_profile.xml file
 */
class UserProfileActivity : BaseActivity(){

    lateinit var gbloggedUser : UserClass

    val firestore = Firebase.firestore

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_user_profile)
        hideStatusBar()

        setupActionBar()
    }

```

```

        populateProfileActivity()

        bt_change_pass.setOnClickListener{
            changePass()
        }

    }

//CUSTOM ACTION BAR
private fun setupActionBar() {
    setSupportActionBar(toolbar_profile)
    val actionBar = supportActionBar
    actionBar!!.setDisplayHomeAsUpEnabled(true)
    actionBar.setHomeAsUpIndicator(R.drawable.ic_back)
    actionBar.title = "Profile"

    toolbar_profile.setNavigationOnClickListener{
        onBackPressed() }
    }

// POPULATE THE PROFILE PAGE WITH USER DATA FROM DB
fun populateProfileActivity () {
    firestore.collection(Constants.USERS)
        .document(getCurrentUserID()).get()
        .addOnSuccessListener {documentSnapshot ->
            Log.d(ContentValues.TAG, "Document received")
            val loggedUser: UserClass? = documentSnapshot.toObject(UserClass::class.java)
            if (loggedUser != null) {
                gbloggedUser = loggedUser
            }
        }
    }
}

```

```

        tv_username_profile.text = loggedUser.name

        tv_profile_name.text=loggedUser.name

        tv_profile_email.text = loggedUser.email

        tv_score.text = ""
    }
}

.addOnFailureListener{
    exception ->
    Log.d(ContentValues.TAG, "get failed with ", exception)
}

}

```

//Display a custom dialog designed in dialog_change_pass that allows users to change their password

// the new entered password will be update don the authentication server

```

private fun changePass(){
    val changePassDialogView = LayoutInflater.from(this).inflate(R.layout.dialog_change_pass,null)
    val changePassDialogBuilder = AlertDialog.Builder(this).setView(changePassDialogView)
    val changePassDialog = changePassDialogBuilder.show()
    changePassDialogView.bt_submit.setOnClickListener{
        val password = changePassDialogView.et_pass.text.toString()
        val reTypedPassword = changePassDialogView.et_rePass.text.toString()
        if(isPasswordMatched(password,reTypedPassword)){

```

```

FirebaseAuth.getInstance().currentUser!!.updatePassword(password).addOnCompleteListener{
    task ->
    if(task.isSuccessful){
        Toast.makeText(this, "Password changed Successfully",
            Toast.LENGTH_SHORT).show()
        changePassDialog.dismiss()
    }else{

```

```

        Toast.makeText(this,task.exception!!.message.toString(),Toast.LENGTH_LONG).show()
    }
}
}else{
    Toast.makeText(this, "Passwords do not match",
        Toast.LENGTH_SHORT).show()
    }
}

changePassDialog.bt_cancel.setOnClickListener{
    changePassDialog.cancel()
}
}

//checks if both the passwords matches
private fun isPasswordMatched (password:String, reTypedPassword:String) :Boolean{
    return password == reTypedPassword
}
}

/**
 * the Adapter class of Activity list RecyclerView
 * source followed: https://developer.android.com/guide/topics/ui/layout/recyclerview
 * https://www.youtube.com/watch?v=6Gm3eMG8KqI
 * https://www.youtube.com/watch?v=afl\_i6uvvU0
 * https://guides.codepath.com/android/using-the-recyclerview
 */
class ActivityAdapter (private val list : ArrayList<ActivityClass> )
:RecyclerView.Adapter<RecyclerView.ViewHolder>(){

    private var onItemClickListener : OnItemClickListener? = null

```



```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {  
    return ActivitiesViewHolder(LayoutInflater.from(parent.context)  
        .inflate(R.layout.item_activity, parent, false))  
}
```

```
override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {  
    val item = list[position]  
    if (holder is ActivitiesViewHolder){  
        holder.itemView.tv_activityName.text=item.name  
        holder.itemView.tv_date.text = item.date  
        holder.itemView.setOnClickListener{  
            if(onItemClickListener!=null){  
                onItemClickListener!!.onItemClick(position,item)  
            }  
        }  
    }  
}
```

```
override fun getItemCount(): Int {  
    return list.size  
}
```

```
class ActivitiesViewHolder(itemView:View) : RecyclerView.ViewHolder(itemView){  
    val textView1 : TextView = itemView.tv_activityName  
    val textView2 : TextView = itemView.tv_date  
}
```

```
fun setOnClickListener(onClickListener: OnItemClickListener){  
    this.onItemClickListener = onItemClickListener  
}
```

```

interface OnItemClickListener {

    fun onItemClick(position: Int, item: ActivityClass)

}

}

/**
 * the Adapter class of Record list RecyclerView
 * source followed: https://developer.android.com/guide/topics/ui/layout/recyclerview
 * https://www.youtube.com/watch?v=6Gm3eMG8KqI
 * https://www.youtube.com/watch?v=afl\_i6uvvU0
 * https://guides.codepath.com/android/using-the-recyclerview
 */

class RecordAdapter (private val list : ArrayList<ActivityRecordClass> ) :
    RecyclerView.Adapter<RecyclerView.ViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {

        return RecordViewHolder(

            LayoutInflater.from(parent.context)

                .inflate(R.layout.item_details, parent, false)

        )

    }

    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {

        val item = list[position]

        if (holder is RecordViewHolder){

            holder.itemView.tv_date.text = item.date

            holder.itemView.tv_timeSpent.text= formatProgress(item.progress)

        }

    }

}

```

```

    }
}

override fun getItemCount(): Int {
    return list.size
}

class RecordViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){

}

private fun formatProgress(progressInSec : Long) : String{
    val hours = progressInSec / 3600
    val minutes = (progressInSec % 3600) / 60;
    val seconds = progressInSec % 60;

    return String.format("%02d:%02d:%02d", hours, minutes, seconds);
}
}

/**
 * Constant variable used though out the development
 */
object Constants {

    const val USERS: String ="Users"
    const val ACTIVITIES : String = "activities"
    const val RECORDS: String = "records"
    const val ACTIVITYNAME : String = "activityName"
}

```

```

/**
 * A data class to hold the Activity details
 */
data class ActivityClass(
    val name: String = "",
    val date : String = SimpleDateFormat("dd/MM/yyyy",Locale.getDefault()).format(Date()),
    val description: String = "",
    val dueDate : String = ""

    ) : Parcelable {
constructor(parcel: Parcel) : this(
    parcel.readString()!!,
    parcel.readString()!!,
    parcel.readString()!!,
    parcel.readString()!!) {
}

override fun writeToParcel(parcel: Parcel, flags: Int) {
    parcel.writeString(name)
    parcel.writeString(date)
    parcel.writeString(description)
    parcel.writeString(dueDate)
}

override fun describeContents(): Int {
    return 0
}

companion object CREATOR : Parcelable.Creator<ActivityClass> {
    override fun createFromParcel(parcel: Parcel): ActivityClass {

```

```

        return ActivityClass(parcel)
    }

    override fun newArray(size: Int): Array<ActivityClass?> {
        return arrayOfNulls(size)
    }
}

/**
 * A data class to hold the record details
 */

data class ActivityRecordClass (
    val progress : Long = 0,
    val date : String = SimpleDateFormat("dd/MM/yyyy", Locale.getDefault()).format(Date())
) : Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readLong(),
        parcel.readString()!!) {
    }

    override fun writeToParcel(parcel: Parcel, flags: Int) {
        parcel.writeLong(progress)
        parcel.writeString(date)
    }

    override fun describeContents(): Int {
        return 0
    }
}

```

```

companion object CREATOR : Parcelable.Creator<ActivityRecordClass> {
    override fun createFromParcel(parcel: Parcel): ActivityRecordClass {
        return ActivityRecordClass(parcel)
    }

    override fun newArray(size: Int): Array<ActivityRecordClass?> {
        return arrayOfNulls(size)
    }
}

/**
 * All the methods of storing the data on the firestore were implemented in this class
 * note: retrieving data methods were not implemented in this file
 * resources used: https://firebase.google.com/docs/auth/android/password-auth
 * https://firebase.google.com/docs/firestore/quickstart#android
 * https://firebase.google.com/docs/auth/android/manage-users
 * https://www.youtube.com/watch?v=dRYnm\_k3w1w&t=269s
 * https://firebase.google.com/docs/firestore/manage-data/add-data
 */
class FirestoreClass {
    val firestore = Firebase.firestore
    private val auth = FirebaseAuth.getInstance()

    //store the user info on the firestore
    fun registerUserOnDB (user: UserClass){
        firestore.collection(Constants.USERS).document(getCurrentUserID())
            .set(user, SetOptions.merge()).addOnSuccessListener {
                Log.d("DDDBB", "Document saved")
            }
    }
}

```

```

    }

    .addOnFailureListener{ e->
        Log.w("DDDBB", "Error adding document")
    }
}

```

// get the ID of currently logged in user from Authentication server

```

fun getCurrentUserID() : String {
    var currentUser = auth.currentUser
    var currentUserID= ""
    if(currentUser != null){
        currentUserID =currentUser.uid
    }
    return currentUserID
}

```

// this function takes a Activity object class and store it in the Firestore as a document in the Activity collection

```

fun saveActivityOnDB(activity: ActivityClass) {
    firestore.collection(Constants.USERS)
        .document(getCurrentUserID())
        .collection(Constants.ACTIVITIES).document(activity.name).set(activity,
SetOptions.merge())
        .addOnSuccessListener {
            Log.d("DDDBB", "Activity saved")
        }
        .addOnFailureListener{ e ->
            Log.e("DDDBB", "activity not saved")
        }
}
}

```

// store the activity record on the firestore

```

fun saveRecordOnDB(record: ActivityRecordClass, activityName :String){
    firestore.collection(Constants.USERS)
        .document(getCurrentUserID())
        .collection(Constants.ACTIVITIES)
        .document(activityName).collection(Constants.RECORDS)
        .add(record).addOnSuccessListener {
            Log.d("DDDBB", "record saved")
        }
        .addOnFailureListener{ e ->
            Log.e("DDDBB", "record not saved")
        }
    }
}

/**
 * A data class to hold the User details
 */
data class UserClass (
    val id: String = "",
    val name : String="",
    val email :String = "",
    val score : Long = 0

    ) : Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readString()!!,
        parcel.readString()!!,
        parcel.readString()!!,
        parcel.readLong()
    ){
    }
}

```



```
override fun writeToParcel(parcel: Parcel, flags: Int) {  
    parcel.writeString(id)  
    parcel.writeString(name)  
    parcel.writeString(email)  
    parcel.writeLong(score)  
}  
  
override fun describeContents(): Int {  
    return 0  
}  
  
companion object CREATOR : Parcelable.Creator<UserClass> {  
    override fun createFromParcel(parcel: Parcel): UserClass {  
        return UserClass(parcel)  
    }  
  
    override fun newArray(size: Int): Array<UserClass?> {  
        return arrayOfNulls(size)  
    }  
}  
}
```

Source code: XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".activities.ActivitiesActivity">
    <include layout="@layout/app_bar2"/>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_activitiesList"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="5dp"
        tools:listitem="@layout/item_activity"
        />

</LinearLayout>
```

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".activities.AddActivity">
```

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    >
```

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar_addActivity"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="#6C4FF1"
    />
```

```
</com.google.android.material.appbar.AppBarLayout>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:orientation="vertical">
```

```
<EditText
    android:id="@+id/et_activityName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Activity name"
```

```
android:inputType="text"
```

```
android:textSize="20sp"
```

```
/>
```

```
<EditText
```

```
android:id="@+id/et_description"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:hint="description"
```

```
android:inputType="text"
```

```
android:textSize="20sp"
```

```
/>
```

```
<EditText
```

```
android:id="@+id/et_dueDate"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:hint="due date"
```

```
android:inputType="date"
```

```
android:textSize="20sp"
```

```
/>
```

```
</LinearLayout>
```

```
<android.widget.Button
```

```
android:id="@+id/bt_add"
```

```
android:layout_gravity="center"
```

```
android:layout_width="200dp"
```

```
android:layout_height="wrap_content"
```

```
android:layout_margin="5dp"
```

```
android:background="@drawable/my_button"
```

```
android:padding="15dp"
```

```

        android:gravity="center"

        android:text="add"

        android:textColor="@color/white"

    />
</LinearLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.ChartsActivity"
    android:orientation="vertical">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="15dp"
        >

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar_charts"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="#6C4FF1"
            />
    </com.google.android.material.appbar.AppBarLayout>

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```
    android:layout_marginTop="2dp"
    android:layout_marginRight="8dp"
    android:layout_marginLeft="8dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:text="Pie Chart"
            android:textColor="@color/black"
        />
        <com.github.mikephil.charting.charts.PieChart
            android:id="@+id/crt_pieChart"
            android:layout_width="match_parent"
            android:layout_height="250dp"/>
    </LinearLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginRight="8dp"
    android:layout_marginLeft="8dp">
    <LinearLayout
        android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:text="Bar Chart"
            android:textColor="@color/black"
        />
        <com.github.mikephil.charting.charts.BarChart
            android:id="@+id/crt_barChart"
            android:layout_width="match_parent"
            android:layout_height="250dp"/>
    </LinearLayout>

```

```

</androidx.cardview.widget.CardView>

```

```

</LinearLayout>

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.DetailsActivity"
    android:orientation="vertical">
    <include layout="@layout/app_bar3"/>

```

```

<LinearLayout

```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center">
        <TextView
            android:id="@+id/tv_activityName"
            android:layout_marginTop="10dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            tools:text="name"
            android:textColor="@color/black"
            android:textSize="20sp"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        >
        <TextView
            android:id="@+id/tv_description"
            android:layout_marginTop="10dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            tools:text="Description"
            android:textColor="@color/black"
            android:textSize="15sp"/>

    </LinearLayout>

    <LinearLayout
```



```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_margin="4dp"
    >
    <TextView
        android:id="@+id/tv_startedDate"
        android:layout_marginTop="10dp"
        android:layout_width="wrap_content"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        tools:text="started date:"
        android:textSize="15sp"/>
```

```
    <TextView
        android:id="@+id/tv_dueDate"
        android:layout_marginTop="10dp"
        android:layout_width="wrap_content"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        tools:text="Due date:"
        android:textSize="15sp"/>
```

```
</LinearLayout>
```

```
<TextView
    android:id="@+id/tv_progress"
    android:layout_marginTop="10dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:text="Time spent:"
```

```
android:textSize="15sp"/>
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:padding="15dp"  
    >  
    <androidx.recyclerview.widget.RecyclerView  
        android:id="@+id/rv_recordList"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent" />  
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".activities.IntroActivity"  
    android:gravity="center_horizontal"  
    android:orientation="vertical"  
    android:background="@drawable/ic_bg"  
    >
```

```
<ImageView  
    android:id="@+id/iv_picstatictic"  
    android:layout_width="531dp"  
    android:layout_height="444dp"
```

```
        android:src="@drawable/picstatistic" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Tractivity"
    android:layout_marginTop="0dp"

    android:textStyle="bold"
    android:textSize="40sp"
    android:textColor="@android:color/white"

/>
```

```
<TextView
    android:id="@+id/tv_welcome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:gravity="center"
    android:text="Welcome"
    android:textColor="#363A43"
    android:textSize="30sp"
    android:textStyle="bold" />
```

```
<TextView
    android:id="@+id/tv_description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="40dp"
    android:gravity="center"
```

```
    android:text="An app to boost productivity"
    android:textColor="@color/white"
    android:textSize="20sp"
  />
```

```
<LinearLayout
```

```
    android:id="@+id/l_buttons"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

```
<android.widget.Button
```

```
    android:id="@+id/bt_login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/my_button"
    android:text="Login"
    android:layout_weight="1"
    android:padding="10dp"
    android:textColor="@color/white"
    android:layout_margin="5dp"
```

```
/>
```

```
<android.widget.Button
```

```
    android:id="@+id/bt_signUp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/my_button"
    android:layout_weight="1"
    android:text="Sign Up"
    android:padding="10dp"
```

```
        android:textColor="@color/white"
        android:layout_margin="5dp"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.LoginActivity"
    android:background="@drawable/ic_bg"
    android:fitsSystemWindows="true"
```

```
>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
```

```
>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tractivity"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="50dp"
```

```
android:layout_marginLeft="10dp"
android:textStyle="bold"
android:textSize="40sp"
android:textColor="@android:color/white">
```

```
</TextView>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="100dp"
    android:text="Welcome Back"
    android:textColor="#363A43"
    android:textSize="30sp"
    android:textStyle="bold"
    android:layout_marginStart="10dp" />
```

```
<EditText
    android:id="@+id/et_email_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="@drawable/edit_text_background"
    android:drawablePadding="10dp"
    android:ems="10"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:padding="10dp"
    android:textColor="@color/white"
```

```
android:textColorHint="@color/white"
android:textSize="15sp"
android:drawableRight="@drawable/ic_email" />
```

```
<EditText
```

```
    android:id="@+id/et_password_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="@drawable/edit_text_background"
    android:drawablePadding="10dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="10dp"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="15sp"
    android:drawableRight="@drawable/ic_password" />
```

```
<TextView
```

```
    android:id="@+id/tv_forgot_pass"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="50dp"
    android:textSize="15sp"
    android:gravity="right"
    android:text="@string/forgot_pass"
    android:textColor="@color/white" />
```

```
<android.widget.Button
```

```
    android:id="@+id/bt_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:background="@drawable/my_button"
    android:padding="15dp"
    android:gravity="center"
    android:text="Login"
    android:textColor="@color/white"

/>
```

```
<TextView
    android:id="@+id/tv_signUP"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/sign_up"
    android:textColor="@color/white"
    android:gravity="center"
    android:layout_marginTop="20dp">
```

```
</TextView>
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```



```
tools:context=".activities.PasswordResetActivity"
android:background="@drawable/ic_bg"
android:fitsSystemWindows="true"
```

```
>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
```

```
>
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tractivity"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="50dp"
    android:layout_marginLeft="10dp"
    android:textStyle="bold"
    android:textSize="40sp"
    android:textColor="@android:color/white">
```

```
</TextView>
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="100dp"
```

```
android:text="Reset your Password"
android:textColor="#363A43"
android:textSize="30sp"
android:textStyle="bold"
android:layout_marginStart="10dp" />
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:textSize="15sp"
    android:gravity="center"
    android:text="Please enter your email address, a email will be set to reset your password"
    android:textColor="@color/white" />
```

```
<EditText
```

```
    android:id="@+id/et_email_passReset"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:background="@drawable/edit_text_background"
    android:drawablePadding="10dp"
    android:ems="10"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:padding="10dp"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="15sp"
    android:drawableRight="@drawable/ic_email" />
```

```
<android.widget.Button
    android:id="@+id/btn_resetPass"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:background="@drawable/my_button"
    android:padding="15dp"
    android:gravity="center"
    android:text="Submit"
    android:textColor="@color/white"

/>
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.SignUpActivity"
    android:background="@drawable/ic_bg"
    android:fitsSystemWindows="true"

>
```

```
<LinearLayout
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tractivity"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="50dp"
    android:layout_marginLeft="10dp"
    android:textStyle="bold"
    android:textSize="40sp"
    android:textColor="@android:color/white">
```

```
</TextView>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="50dp"
    android:text="Sign up"
    android:textColor="#363A43"
    android:textSize="30sp"
    android:textStyle="bold"
    android:paddingLeft="10dp"
    android:paddingStart="10dp" />
```

```
<EditText
    android:id="@+id/et_name"
    android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:background="@drawable/edit_text_background"
android:drawablePadding="10dp"
android:ems="10"
android:hint="Username"
android:inputType="textPersonName"
android:padding="10dp"
android:textColor="@color/white"
android:textColorHint="@color/white"
android:textSize="15sp"
android:drawableRight="@drawable/ic_user" />
```

<EditText

```
android:id="@+id/et_email"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:background="@drawable/edit_text_background"
android:drawablePadding="10dp"
android:ems="10"
android:hint="Email"
android:inputType="textEmailAddress"
android:padding="10dp"
android:textColor="@color/white"
android:textColorHint="@color/white"
android:textSize="15sp"
android:drawableRight="@drawable/ic_email" />
```

<EditText

```
android:id="@+id/et_password"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:background="@drawable/edit_text_background"
    android:drawablePadding="10dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="10dp"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="15sp"
    android:drawableRight="@drawable/ic_password" />
<EditText
    android:id="@+id/et_reTypePassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="50dp"
    android:background="@drawable/edit_text_background"
    android:drawablePadding="10dp"
    android:ems="10"
    android:hint="Re-enter password"
    android:inputType="textPassword"
    android:padding="10dp"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="15sp"
    android:drawableRight="@drawable/ic_password" />
```

```
<android.widget.Button
    android:id="@+id/bt_SignUp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:background="@drawable/my_button"
    android:padding="15dp"
    android:gravity="center"
    android:text="Sign Up"
    android:textColor="@color/white"

/>
```

```
<TextView
    android:id="@+id/tv_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/login"
    android:textColor="@color/white"
    android:gravity="center"
    android:layout_marginTop="20dp"

>
```

```
</TextView>
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/ic_bg"
tools:context=".activities.TractivityMain">
```

```
<ImageView

    android:id="@+id/bg_Clock"

    android:layout_width="317dp"
    android:layout_height="294dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:src="@drawable/clock"

    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.469"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<ImageView

    android:id="@+id/iArrow"

    android:layout_width="285dp"
    android:layout_height="229dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
```



```
android:layout_marginTop="116dp"
android:layout_marginEnd="8dp"
android:layout_marginRight="8dp"
android:foregroundGravity="center"
android:src="@drawable/arrow_red"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.465"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<android.widget.Button
    android:id="@+id/btn_start"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:layout_marginStart="68dp"
    android:layout_marginLeft="68dp"
    android:layout_marginTop="49dp"
    android:layout_marginBottom="240dp"
    android:background="@drawable/button_start"
    android:text="Start"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/bg_Clock"
    app:layout_constraintVertical_bias="1.0"
    tools:ignore="MissingConstraints" />
```

```
<android.widget.Button
    android:id="@+id/btn_stop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
android:layout_marginTop="49dp"
android:layout_marginEnd="40dp"
android:layout_marginRight="40dp"
android:layout_marginBottom="240dp"
android:background="@drawable/button_stopwatch"
android:text="Stop"
android:textColor="@color/white"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.756"
app:layout_constraintStart_toEndOf="@+id/btn_start"
app:layout_constraintTop_toBottomOf="@+id/bg_Clock"
app:layout_constraintVertical_bias="1.0"
tools:ignore="MissingConstraints" />
```

<Chronometer

```
android:id="@+id/c_chronometer"
android:layout_width="117dp"
android:layout_height="58dp"
android:layout_marginStart="40dp"
android:layout_marginLeft="40dp"
android:layout_marginEnd="40dp"
android:layout_marginRight="40dp"
android:layout_marginBottom="308dp"
android:gravity="center"
android:text="TextView"
android:textColor="@color/black"
android:textSize="30sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.488"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/bg_Clock"
app:layout_constraintVertical_bias="0.0"
tools:ignore="MissingConstraints" />
```

```
<Button
    android:id="@+id/btn_reset"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/button_stopwatch"
    android:text="Reset"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/c_chronometer"
    app:layout_constraintVertical_bias="0.326" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">
```

```
<include
    layout="@layout/app_bar"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

<com.google.android.material.navigation.NavigationView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:id="@+id/navigation"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header"
    app:menu="@menu/drawer_menu" />
```

```
</androidx.drawerlayout.widget.DrawerLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".activities.UserProfileActivity"
    android:gravity="top"
    android:orientation="vertical">
```

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    >
```

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar_profile"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="#6C4FF1"
/>
```

```
</com.google.android.material.appbar.AppBarLayout>
```

```
<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/profile_image"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:src="@drawable/profile_pic"
    app:civ_border_color="#FF000000"
    app:civ_border_width="2dp"
/>
```

```
<TextView
    android:id="@+id/tv_username_profile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_gravity="center"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    tools:text="Username"
```

```
    android:gravity="center"
    android:layout_marginBottom="20dp"
  />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="center"
    android:paddingLeft="15dp"

    >
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    android:gravity="center"
    android:text="Name:"
    android:layout_marginRight="10dp"
  />
```

```
<TextView
    android:id="@+id/tv_profile_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    tools:text="name"
```

```
/>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_gravity="center"  
    android:paddingLeft="15dp"
```

```
>
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="20dp"  
    android:textColor="@android:color/black"  
    android:textSize="20sp"  
    android:gravity="center"  
    android:text="Email:"  
    android:layout_marginRight="10dp"  
/>
```

```
<TextView
```

```
    android:id="@+id/tv_profile_email"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="20dp"  
    android:textColor="@android:color/black"  
    android:textSize="20sp"
```

```
tools:text="example@email.com"
android:text=""
/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="center"
    android:paddingLeft="15dp"

    >
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    android:gravity="center"
    android:text="Productivity Score:"
    android:layout_marginRight="10dp"
/>
```

```
<TextView
    android:id="@+id/tv_score"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
```



```
    android:textColor="@android:color/black"
    android:textSize="20sp"
    tools:text="0"
/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="center"
    android:paddingLeft="15dp"
```

```
>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    android:gravity="center"
    android:text="Mobile:"
    android:layout_marginRight="10dp"
/>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
```

```
    android:textColor="@android:color/black"
    android:textSize="20sp"
    tools:text="07123145678"
/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="center"
    android:paddingLeft="15dp"

>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    android:gravity="center"
    android:text="Address:"
    android:layout_marginRight="10dp"
/>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="20dp"
    android:textColor="@android:color/black"
    android:textSize="20sp"
    tools:text="address"
/>
```

```
</LinearLayout>
```

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/bt_change_pass"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="50dp"
    android:text="Change password"
/>
```

```
</LinearLayout>
```

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
>
```

```
<androidx.appcompat.widget.Toolbar
```

```
    android:id="@+id/toolbar_trmain"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="#6C4FF1"
/>
```

```
</com.google.android.material.appbar.AppBarLayout>
```

```
<include layout="@layout/activity_stopwatch" />
```

```
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

```
<RelativeLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#6C4FF1">
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:padding="16dp">
```

```
<ImageView
```

```
    android:id="@+id/iv_back"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_back"
/>
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_toEndOf="@+id/iv_back"
    android:layout_marginStart="16dp"
    android:text="Activities"
    android:textSize="20sp"
    android:layout_toRightOf="@+id/iv_back"
    android:layout_marginLeft="15dp" />
```

```
<ImageView
```

```
    android:id="@+id/iv_add"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_add"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true" />
```

```
</RelativeLayout>
```

```
</RelativeLayout>
```

```
<RelativeLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#6C4FF1">
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:padding="16dp">
```

```
<ImageView
```

```

        android:id="@+id/iv_back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_back"
    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toEndOf="@+id/iv_back"
    android:layout_marginStart="16dp"
    android:text="Activity Details"
    android:textSize="20sp"
    android:layout_toRightOf="@+id/iv_back"
    android:layout_marginLeft="15dp" />
<ImageView
    android:id="@+id/iv_edit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_edit"
    android:layout_toLeftOf="@+id/iv_delete"
    android:layout_marginRight="20dp"
    />
<ImageView
    android:id="@+id/iv_delete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_delete"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true" />

</RelativeLayout>

```

```
</RelativeLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical">
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Change Password"  
    android:textColor="@android:color/black"  
    android:textSize="26sp"  
    android:layout_gravity="center"/>
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="8dp"  
    android:orientation="vertical">
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_margin="10dp">  
    <TextView  
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_weight="3"
android:text="Enter new password"
android:textSize="15sp"
android:paddingBottom="10dp"/>
```

```
<EditText
```

```
    android:id="@+id/et_pass"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:inputType="textPassword"
    android:textSize="15sp"
    android:paddingTop="10dp"
    />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_margin="10dp">
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="3"
    android:text="Re-enter password"
    android:textSize="15sp"
    android:paddingBottom="10dp"/>
```

```
<EditText
```



```
        android:id="@+id/et_rePass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:inputType="textPassword"
        android:textSize="15sp"
    />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">
```

```
<androidx.appcompat.widget.AppCompatButton
```

```
    android:id="@+id/bt_cancel"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:foreground="?attr/selectableItemBackground"
    android:gravity="center"
    android:padding="10dp"
    android:text="CANCEL"
    android:textColor="@android:color/black"/>
```

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/bt_submit"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:foreground="?attr/selectableItemBackground"
    android:gravity="center"
    android:padding="10dp"
    android:text="SUBMIT"
    android:textColor="@android:color/holo_red_dark"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Edit Activity"
    android:textColor="@android:color/black"
    android:textSize="26sp"
    android:layout_gravity="center"/>
```

```
<LinearLayout
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="8dp"
android:orientation="vertical">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

```
    <TextView
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:text="Description:"
        android:textSize="15sp"/>
```

```
    <EditText
```

```
        android:id="@+id/et_description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:inputType="text"
        android:textSize="15sp"
    />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:orientation="horizontal">
```

```
<TextView
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_weight="3"
```

```
    android:text="Due Date:"
```

```
    android:textSize="15sp"/>
```

```
<EditText
```

```
    android:id="@+id/et_dueDate"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_weight="1"
```

```
    android:inputType="date"
```

```
    android:textSize="15sp"
```

```
/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="10dp"
```

```
    android:orientation="horizontal">
```

```
<androidx.appcompat.widget.AppCompatButton
```

```
    android:id="@+id/bt_cancel"
```

```
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:foreground="?attr/selectableItemBackground"
    android:gravity="center"
    android:padding="10dp"
    android:text="CANCEL"
    android:textColor="@android:color/black"/>
```

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/bt_submit"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:foreground="?attr/selectableItemBackground"
    android:gravity="center"
    android:padding="10dp"
    android:text="SUBMIT"
    android:textColor="@android:color/holo_red_dark"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical">
```

```
<TextView
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Save Activity"
android:textColor="@android:color/black"
android:textSize="26sp"
android:layout_gravity="center"/>
```

```
<TextView
```

```
    android:id="@+id/tv_description"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    tools:text="time spent:"
    android:textColor="@android:color/darker_gray"
    android:textSize="22sp" />
```

```
<EditText
```

```
    android:id="@+id/et_activityName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Activity name"
    android:inputType="text"
    android:textSize="20sp"
    />
```

```
<androidx.appcompat.widget.AppCompatButton
```

```
    android:id="@+id/bt_selectActivity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Or Select Activity"
```

```
/>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="10dp"
```

```
    android:orientation="horizontal">
```

```
<androidx.appcompat.widget.AppCompatButton
```

```
    android:id="@+id/bt_cancel"
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_weight="1"
```

```
    android:foreground="?attr/selectableItemBackground"
```

```
    android:gravity="center"
```

```
    android:padding="10dp"
```

```
    android:text="CANCEL"
```

```
    android:textColor="@android:color/black"/>
```

```
<androidx.appcompat.widget.AppCompatButton
```

```
    android:id="@+id/bt_submit"
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_weight="1"
```

```
    android:foreground="?attr/selectableItemBackground"
```

```
    android:gravity="center"
```

```
    android:padding="10dp"
```

```
    android:text="SUBMIT"
```

```
    android:textColor="@android:color/holo_red_dark"/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<androidx.cardview.widget.CardView  
xmlns:android="http://schemas.android.com/apk/res/android"  
  
xmlns:tools="http://schemas.android.com/tools"  
  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_margin="5dp">
```

```
<RelativeLayout  
  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:padding="5dp">
```

```
<TextView  
  
    android:id="@+id/tv_activityName"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    tools:text="activity name"  
    android:textColor="@color/black"  
    android:textSize="20sp"  
    android:textStyle="bold" />
```

```
<TextView  
  
    android:id="@+id/tv_startedOn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```



```
android:layout_below="@id/tv_activityName"
android:text="started on:" />
```

```
<TextView
    android:id="@+id/tv_date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/tv_activityName"
    android:layout_toEndOf="@id/tv_startedOn"
    android:layout_toRightOf="@id/tv_startedOn"
    android:paddingStart="3dp"
    android:paddingLeft="3dp"
    tools:text="dd/mm/yyyy" />
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="2dp">
```

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="5dp">
```

```
<TextView
    android:id="@+id/tv_date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:text="date"
    android:textColor="@color/black"
    android:textStyle="bold" />
```

```
<TextView
    android:id="@+id/tv_startedOn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/tv_date"
    android:text="Time spent:" />
```

```
<TextView
    android:id="@+id/tv_timeSpent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/tv_date"
    android:layout_toEndOf="@id/tv_startedOn"
    android:layout_toRightOf="@id/tv_startedOn"
    android:paddingStart="3dp"
    android:paddingLeft="3dp"
    tools:text="1hr" />
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="200dp"
xmlns:tools="http://schemas.android.com/tools"
android:background="@drawable/nav_header_background"
android:gravity="bottom"
android:orientation="vertical"
```

```
>
```

```
<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/profile_image"
    android:layout_width="96dp"
    android:layout_height="96dp"
    android:layout_marginStart="15dp"
    android:layout_marginEnd="15dp"
    android:layout_gravity="bottom"
    android:layout_marginLeft="15dp"
    android:src="@drawable/profile_pic"
    app:civ_border_color="#FF000000"
    app:civ_border_width="2dp" />
```

```
<TextView
    android:id="@+id/tv_username"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="10dp"
    android:textColor="@android:color/white"
    android:textSize="15sp"
```

```
tools:text="Username"
```

```
android:layout_marginLeft="15dp" />
```

```
</LinearLayout>
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<item
```

```
    android:id="@+id/main_pg"
```

```
    android:title="Tractivity"
```

```
    android:icon="@drawable/ic_stopwatch"/>
```

```
<item
```

```
    android:id="@+id/activities"
```

```
    android:title="Activities"
```

```
    android:icon="@drawable/ic_activities"
```

```
/>
```

```
<item
```

```
    android:id="@+id/charts"
```

```
    android:title="Charts"
```

```
    android:icon="@drawable/ic_charts"
```

```
/>
```

```
<item
```

```
    android:id="@+id/profile"
```

```
    android:title="Profile"
```

```
    android:icon="@drawable/ic_profile"
```

```
/>
```

```
<item
```

```
        android:id="@+id/logout"
        android:title="Logout"
        android:icon="@drawable/ic_logout"
    />
</menu>
```