Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт «**Практическая работа №3**»

Выполнила: Зиятдинова Алина Ленаровна

Группа: ПР-31

Преподаватель:

2023

|  |
| --- |
| **Manifest** |
| <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android">   <uses-permission android:name="android.permission.INTERNET" />  <uses-feature  android:name="android.software.leanback"  android:required="false" />  <uses-feature  android:name="android.hardware.touchscreen"  android:required="false" />   <application  android:allowBackup="true"  android:banner="@drawable/banner"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/Theme.AppCompat.NoActionBar">  <activity  android:name=".QuestsActivity"  android:exported="false" />  <activity  android:name=".SignInActivity"  android:exported="false" />  <activity android:name=".MovieDetailsActivity"  android:exported="false"/>  <activity  android:name=".MainActivity"  android:exported="true"  android:label="@string/app\_name"  android:screenOrientation="landscape">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LEANBACK\_LAUNCHER" />  </intent-filter>  </activity>  </application> </manifest> |

|  |
| --- |
| **build.gradle** |
| plugins **{** id("com.android.application")  id("org.jetbrains.kotlin.android") **}** *android* **{** namespace = "com.example.tvapplication"  compileSdk = 34   defaultConfig **{** applicationId = "com.example.tvapplication"  minSdk = 28  targetSdk = 33  versionCode = 1  versionName = "1.0"   **}** buildFeatures**{** viewBinding = true  **}** buildTypes **{** *release* **{** isMinifyEnabled = false  proguardFiles(  getDefaultProguardFile("proguard-android-optimize.txt"),  "proguard-rules.pro"  )  **}  }** compileOptions **{** sourceCompatibility = JavaVersion.*VERSION\_1\_8* targetCompatibility = JavaVersion.*VERSION\_1\_8* **}** *kotlinOptions* **{** jvmTarget = "1.8"  **} }** *dependencies* **{** *implementation*("androidx.core:core-ktx:1.12.0")  *implementation*("androidx.leanback:leanback:1.0.0")  *implementation*("androidx.appcompat:appcompat:1.6.1")  *implementation*("androidx.media2:media2-widget:1.2.1")  *implementation*("com.google.android.material:material:1.10.0")  *implementation*("androidx.constraintlayout:constraintlayout:2.1.4")  *implementation*("com.android.volley:volley:1.2.1")  *implementation*("com.squareup.picasso:picasso:2.71828")  *implementation*("com.google.code.gson:gson:2.9.0") **}** |

**MainActivity**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <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=".MainActivity"  android:orientation="vertical"  android:background="#bb86fc">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Мобильное TV"  android:textSize="30dp"  android:layout\_gravity="center"  android:textColor="#ffffff"  android:layout\_marginTop="85dp"  android:layout\_marginBottom="25dp"  android:textStyle="bold"  tools:layout\_editor\_absoluteX="134dp"  tools:layout\_editor\_absoluteY="190dp" />  <ImageView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:src="@drawable/hypnosis"  android:layout\_gravity="center" />   </LinearLayout> |
|  |
| class MainActivity : Activity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  val timer = object: CountDownTimer(3000,1000){  override fun onTick(millisUntilFinished: Long) {   }   override fun onFinish() {  startActivity(Intent(this@MainActivity,SignInActivity::class.*java*))  finish()  }  }  timer.start()  } } |

|  |
| --- |
| **Data class Movie** |
| data class Movie(  val title: String,  val description: String,  val posterUrl: String ) |

|  |
| --- |
| **Data class Quests** |
| data class Quests(  val image: Int,  val title: String,  val text: String )  class MyObj(  val list: ArrayList<Quests> = *arrayListOf*(  Quests(  R.drawable.*kazan*,  "Welcome to Kazan",  "Now you must go to Kazan Expo and find next key. And there are many more interesting information"  ),  Quests(  R.drawable.*kazan*,  "Welcome to Kazan",  "Now you must go to Kazan Expo and find next key. And there are many more interesting information"  ),  Quests(  R.drawable.*kazan*,  "Welcome to Kazan",  "Now you must go to Kazan Expo and find next key. And there are many more interesting information"  ),  Quests(  R.drawable.*kazan*,  "Welcome to Kazan",  "Now you must go to Kazan Expo and find next key. And there are many more interesting information"  ),  Quests(  R.drawable.*kazan*,  "Welcome to Kazan",  "Now you must go to Kazan Expo and find next key. And there are many more interesting information"  ),  Quests(  R.drawable.*kazan*,  "Welcome to Kazan",  "Now you must go to Kazan Expo and find next key. And there are many more interesting information"  )  ) ) |

|  |
| --- |
| **MovieAdapter** |
| class MovieAdapter(private val context: Context, private var moviesList: List<Movie>) :  RecyclerView.Adapter<MovieAdapter.MovieViewHolder>(), Filterable {  private var filteredMoviesList: List<Movie> = moviesList.*toList*()   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MovieViewHolder {  val view = LayoutInflater.from(parent.*context*)  .inflate(R.layout.*item\_movie*, parent, false)  return MovieViewHolder(view)  }   override fun onBindViewHolder(holder: MovieViewHolder, position: Int) {  val movie = filteredMoviesList[position]  holder.bind(movie)  }    interface OnItemClickListener {  fun onItemClick(movie: Movie)  }   override fun getFilter(): Filter {  return object : Filter() {  override fun performFiltering(constraint: CharSequence?): FilterResults {  val queryString = constraint?.toString()?.*toLowerCase*()  val filterResults = FilterResults()  filterResults.values = if (queryString.*isNullOrEmpty*()) {  moviesList.*toList*()  } else {  moviesList.*filter* **{  it**.title.*toLowerCase*().*contains*(queryString)  **}** }  return filterResults  }   override fun publishResults(constraint: CharSequence?, results: FilterResults?) {  filteredMoviesList = results?.values as List<Movie>  notifyDataSetChanged()  }  }  }   private var onItemClickListener: OnItemClickListener? = null   fun setOnItemClickListener(listener: OnItemClickListener) {  onItemClickListener = listener  }   override fun getItemCount(): Int = filteredMoviesList.size   inner class MovieViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  private val imageViewPoster: ImageView = itemView.findViewById(R.id.*imageViewPoster*)  private val textViewTitle: TextView = itemView.findViewById(R.id.*textViewTitle*)  private val textViewDescription: TextView = itemView.findViewById(R.id.*textViewDescription*)   init {  itemView.setOnClickListener **{** val position = *adapterPosition* if (position != RecyclerView.*NO\_POSITION*) {  val clickedMovie = moviesList[position]  onItemClickListener?.onItemClick(clickedMovie)  }  **}** }   fun bind(movie: Movie) {  // Установка значений элементов пользовательского интерфейса  textViewTitle.*text* = movie.title  textViewDescription.*text* = movie.description   // Использование библиотеки Picasso для загрузки изображения  Picasso.get()  .load(movie.posterUrl)  .placeholder(R.drawable.*placeholder\_poster*)  .error(R.drawable.*placeholder\_poster*)  .into(imageViewPoster)  }  }   fun setMovies(newMoviesList: List<Movie>) {  moviesList = newMoviesList  filteredMoviesList = newMoviesList  notifyDataSetChanged()  } } |

|  |
| --- |
| **MovieDetailsActivity** |
| class MovieDetailsActivity : Activity() {  private lateinit var binding: ActivityMovieDetailsBinding  private lateinit var requestQueue: RequestQueue   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivityMovieDetailsBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)  val movieTitle = *intent*.getStringExtra("movieTitle")  requestQueue = Volley.newRequestQueue(this)  if (!movieTitle.*isNullOrEmpty*()) {  getMovieDetails(movieTitle)  }  }   private fun getMovieDetails(title: String) {  val apiKey = "bc17ab59"  val url = "https://www.omdbapi.com/?apikey=$apiKey&t=$title"   val request = JsonObjectRequest(  Request.Method.*GET*, url, null,  **{** response **->** try {  Picasso.get()  .load(response.optString("Poster"))  .placeholder(R.drawable.*placeholder\_poster*)  .error(R.drawable.*placeholder\_poster*)  .into(binding.imageViewPosterDetails)  binding.textViewTitle.*text* = title  binding.textViewDescription.*text* = response.optString("Plot")  binding.textViewRating.*text* = "Оценка на IMDB ${response.optString("imdbRating")}"  Log.d("Volley",response.toString())   } catch (e: Exception) {  Log.e("Volley", "Error parsing response: $e")  }  **}**,  **{** error **->** Log.e("Volley", "Error: ${error.message}")  **}** )   requestQueue.add(request)  } } |
| <?xml version="1.0" encoding="utf-8"?> <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_height="wrap\_content"  android:layout\_width="match\_parent">  <LinearLayout  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"  android:orientation="vertical"  android:padding="16dp"  tools:context=".MovieDetailsActivity">   <ImageView  android:id="@+id/imageViewPosterDetails"  android:layout\_width="300dp"  android:layout\_height="300dp"  android:layout\_gravity="center\_horizontal"  android:scaleType="centerCrop" />    <TextView  android:id="@+id/textViewTitle"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="18sp"  android:textStyle="bold"  android:layout\_gravity="center\_horizontal"  android:layout\_marginTop="8dp"  android:layout\_marginBottom="8dp"/>   <TextView  android:id="@+id/textViewDescription"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="16sp"  android:gravity="center\_horizontal"  android:layout\_marginTop="8dp"  android:layout\_marginBottom="8dp"/>   <TextView  android:id="@+id/textViewRating"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="16sp"  android:layout\_gravity="center\_horizontal"  android:layout\_marginTop="8dp"/>  </LinearLayout> </ScrollView> |
|  |
|  |

|  |
| --- |
| **MoviesByGenreFragment** |
| class MoviesByGenreFragment : Fragment() {   private lateinit var binding: FragmentMoviesByGenreBinding  private lateinit var movieAdapter: MovieAdapter  private lateinit var requestQueue: RequestQueue   companion object {  private const val ARG\_KEY = "myKey"   fun newInstance(yourValue: String): MoviesByGenreFragment {  val fragment = MoviesByGenreFragment()  val args = Bundle()  args.putString(ARG\_KEY, yourValue)  fragment.*arguments* = args  return fragment  }  }   override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View {  binding = FragmentMoviesByGenreBinding.inflate(inflater, container, false)  binding.recyclerViewMovies.*layoutManager* = GridLayoutManager(requireContext(), 3)  movieAdapter = MovieAdapter(requireContext(), *emptyList*())  binding.recyclerViewMovies.*adapter* = movieAdapter   movieAdapter.setOnItemClickListener(object : MovieAdapter.OnItemClickListener {  override fun onItemClick(movie: Movie) {  val intent = Intent(requireContext(), MovieDetailsActivity::class.*java*)  intent.putExtra("movieTitle", movie.title)  startActivity(intent)  }  })   val genre = *arguments*?.getString(ARG\_KEY)  requestQueue = Volley.newRequestQueue(requireContext())   if (genre != null) {  fetchMoviesByGenre(genre)  }  binding.searchEditText.addTextChangedListener(object : TextWatcher {  override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {  // Метод, вызываемый перед изменением текста  }   override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {  // Метод, вызываемый при изменении текста  movieAdapter.*filter*.filter(s)  }   override fun afterTextChanged(s: Editable?) {  // Метод, вызываемый после изменения текста  }  })  return binding.*root* }   private fun fetchMoviesByGenre(genre: String) {  val url = "https://www.omdbapi.com/?apikey=bc17ab59&s=${genre}&type=movie"   val request = StringRequest(Request.Method.*GET*, url,  **{** response **->** try {  Log.d("Movie", "Raw JSON Response: $response")   val jsonObject = JSONObject(response)  val error = jsonObject.optString("Error")   if (error.*isNullOrEmpty*()) {  val searchArray = jsonObject.optJSONArray("Search")   // Проверка на null итерируемого объекта  if (searchArray != null) {  val movies = *mutableListOf*<Movie>()   // Вывод списка фильмов  for (i in 0 *until* searchArray.length()) {  val movieObject = searchArray.optJSONObject(i)  val title = movieObject.optString("Title")  val description = movieObject.optString("Plot")  val posterUrl = movieObject.optString("Poster")   movies.add(Movie(title, description, posterUrl))  }   if (movies.*isNotEmpty*()) {  movies.*forEach* **{** Log.d(  "Movie",  "${**it**.title}\n${**it**.description}\n${**it**.posterUrl}\n"  )  **}** movieAdapter.setMovies(movies)  } else {  Log.d("Movie", "No movies found.")  }  } else {  Log.e("Volley", "Search results are null")  }  } else {  Log.e("Volley", "Error: $error")  }  } catch (e: Exception) {  Log.e("Volley", "Error parsing JSON: ${e.message}")  }  **}**,  **{** error **->** Log.e("Volley", "Error: ${error.message}")  **}**)   requestQueue.add(request)  } } |
|  |
| <FrameLayout 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=".MoviesByGenreFragment">   <EditText  android:id="@+id/searchEditText"  android:layout\_width="300dp"  android:layout\_marginTop="20dp"  android:layout\_marginLeft="20dp"  android:background="@drawable/edittext\_design"  android:layout\_height="wrap\_content"  android:layout\_gravity="start"  android:hint="Название" />   <androidx.recyclerview.widget.RecyclerView  android:id="@+id/recyclerViewMovies"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginTop="60dp"  android:clipToPadding="false"  android:padding="8dp"  android:scrollbars="vertical" />  </FrameLayout> |
|  |

|  |
| --- |
| class NewsFragment : Fragment() {  private lateinit var binding:FragmentNewsBinding   override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  binding = FragmentNewsBinding.inflate(inflater,container,false)  binding.recyclers.*layoutManager* = GridLayoutManager(requireContext(),3)  binding.recyclers.*adapter* = QuestsRecycler(requireContext(),MyObj().list)  return binding.*root* } } |
|  |
| <?xml version="1.0" encoding="utf-8"?> <FrameLayout 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=".NewsFragment">   <TextView  android:id="@+id/texts"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif-medium"  android:gravity="center\_horizontal"  android:layout\_marginTop="20dp"  android:text="My quests"  android:textColor="@color/aquamarine"  android:textSize="50sp" />   <androidx.recyclerview.widget.RecyclerView  android:id="@+id/recyclers"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_horizontal"  android:layout\_marginTop="120dp" /> </FrameLayout> |

|  |
| --- |
| **QuestsActivity** |
| class QuestsActivity : AppCompatActivity() {  private lateinit var binding: ActivityQuestsBinding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivityQuestsBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)  binding.bottomNavigationView.setOnNavigationItemSelectedListener **{** item **->** when (item.*itemId*) {  R.id.*news* -> {  loadFragment(NewsFragment())  return@setOnNavigationItemSelectedListener true  }  R.id.*drama* -> {  val selectedValue = "drama"  val fragment = MoviesByGenreFragment.newInstance(selectedValue)  loadFragment(fragment)  return@setOnNavigationItemSelectedListener true  }  R.id.*comedy* -> {  val selectedValue = "comedy"  val fragment = MoviesByGenreFragment.newInstance(selectedValue)  loadFragment(fragment)  return@setOnNavigationItemSelectedListener true  }  R.id.*horror* -> {  val selectedValue = "horror"  val fragment = MoviesByGenreFragment.newInstance(selectedValue)  loadFragment(fragment)  return@setOnNavigationItemSelectedListener true  }  }  false  **}** // По умолчанию загружаем фрагмент "Главная"  loadFragment(NewsFragment())  }   private fun loadFragment(fragment: Fragment) {  *supportFragmentManager*.beginTransaction()  .replace(R.id.*container*, fragment)  .commit()  } } |
|  |
| <?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="@color/background"  tools:context=".QuestsActivity">  <FrameLayout  android:id="@+id/container"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintBottom\_toTopOf="@id/bottomNavigationView"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <com.google.android.material.bottomnavigation.BottomNavigationView  android:id="@+id/bottomNavigationView"  android:layout\_width="0dp"  android:background="#55555555"  android:layout\_height="wrap\_content"  android:layout\_gravity="bottom"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:itemTextColor="@color/white"  style="@style/BottomNavigationStyle"  app:menu="@menu/bottom\_nav\_menu" />  </androidx.constraintlayout.widget.ConstraintLayout> |

|  |
| --- |
| **QuestsRecycler** |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:background="@drawable/gradient"  android:layout\_marginEnd="10dp"  android:layout\_marginBottom="10dp"  android:orientation="vertical">   <ImageView  android:id="@+id/image\_quests"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:src="@drawable/kazan" />  <TextView  android:id="@+id/title\_quest"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="10dp"  android:layout\_marginStart="20dp"  android:textColor="@color/aquamarine"  android:textSize="20sp"  android:fontFamily="sans-serif-medium"  android:text="1111111111111111111111"  app:layout\_constraintBaseline\_toBottomOf="@id/image\_quests"/>  <TextView  android:id="@+id/description"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textColor="@color/white"  android:textSize="20sp"  android:layout\_marginTop="10dp"  android:layout\_marginStart="20dp"  android:layout\_marginEnd="20dp"/>  <TextView  android:id="@+id/bt\_details"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Details"  android:textSize="28sp"  android:textColor="@color/aquamarine"  android:layout\_marginStart="20sp"  android:layout\_marginBottom="20sp"  />  </LinearLayout> |
| class QuestsRecycler(val context: Context, val questsList:ArrayList<Quests>):RecyclerView.Adapter<QuestsRecycler.QuestsViewHolder>() {  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): QuestsViewHolder {  val view = LayoutInflater.from(parent.*context*).inflate(R.layout.*quests\_adapter*, parent, false)  return QuestsViewHolder(view)  }   override fun onBindViewHolder(holder: QuestsViewHolder, position: Int) {  val quest = questsList[position]   holder.imageQuests.setImageResource(quest.image)  holder.titleQuest.*text* = quest.title  holder.textQuest.*text* = quest.text   }   override fun getItemCount(): Int {  return questsList.size  }   inner class QuestsViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  val imageQuests: ImageView = itemView.findViewById(R.id.*image\_quests*)  val titleQuest: TextView = itemView.findViewById(R.id.*title\_quest*)  val textQuest: TextView = itemView.findViewById(R.id.*description*)  val btDetails: TextView = itemView.findViewById(R.id.*bt\_details*)  }  } |

|  |
| --- |
| **SingInActivity** |
| class SignInActivity : Activity() {  lateinit var sharedPreferences: SharedPreferences  lateinit var binding: ActivitySignInBinding  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = ActivitySignInBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)   sharedPreferences = getPreferences(*MODE\_PRIVATE*)  val email = sharedPreferences.getString("login", null)  val password = sharedPreferences.getString("password", null)  if (email != null) {  binding.email.setText(email)  binding.password.setText(password)  }  binding.login.setOnClickListener **{** if (binding.email.*text*.length < 3 || binding.password.*text*.length < 3) {  Toast.makeText(  this,  "Длина логина и пароля должна быть больше 3 символов",  Toast.*LENGTH\_SHORT* ).show()  return@setOnClickListener  }  if(email==null){  sharedPreferences.edit().putString("login",binding.email.*text*.toString()).apply()  sharedPreferences.edit().putString("password",binding.password.*text*.toString()).apply()  startActivity(Intent(this,QuestsActivity::class.*java*))  }  if(binding.email.*text*.toString()==email&&binding.password.*text*.toString()==password)  startActivity(Intent(this,QuestsActivity::class.*java*))  else  Toast.makeText(this,"Неверный логин или пароль",Toast.*LENGTH\_SHORT*).show()  **}** } } |
| <?xml version="1.0" encoding="utf-8"?> <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:background="@drawable/background"  android:gravity="center"  tools:context=".SignInActivity">   <LinearLayout  android:layout\_width="347dp"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:background="@color/background"  android:orientation="vertical"  android:paddingStart="50dp"  android:paddingTop="20dp"  android:paddingEnd="50dp"  android:paddingBottom="20dp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent">   <ImageView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:adjustViewBounds="true"  android:src="@drawable/logo\_ic" />   <EditText  android:id="@+id/email"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="10dp"   android:hint="Email"  android:inputType="textEmailAddress"  android:textColor="@color/aquamarine"  android:textColorHighlight="@color/aquamarine"  android:textColorHint="@color/grey"  android:textCursorDrawable="@drawable/focus\_design"  />   <EditText  android:id="@+id/password"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="10dp"  android:hint="Password"  android:inputType="textPassword"  android:textColor="@color/aquamarine"  android:textColorHighlight="@color/aquamarine"  android:textColorHint="@color/grey"  android:textCursorDrawable="@drawable/focus\_design" />   <Button  android:id="@+id/login"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="30dp"  android:background="@drawable/button\_design"  android:text="Sign in"  android:textAllCaps="false"  android:textSize="18sp" />  </LinearLayout>   </LinearLayout> |
|  |
|  |