```kotlin
package com.example.myapplication

 const val  base_url : String = "https://cptvk1.000webhostapp.com/"
const val  register_url : String = base_url+"register.php"
const val  login_url : String = base_url+"login.php"
const val upload_url : String = base_url+"upload_request.php"
const val get_url : String = base_url+"get_request.php"
```

```kotlin
package com.example.myapplication

import android.Manifest.permission
import android.annotation.SuppressLint
import android.app.Activity
import android.content.ContentUris
import android.content.Intent
import android.database.Cursor
import android.net.Uri
import android.os.Build
import android.os.Bundle
import android.os.Environment
import android.provider.DocumentsContract
import android.provider.MediaStore
import android.view.View
import android.widget.*
import androidx.annotation.RequiresApi
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.PermissionChecker
import androidx.preference.PreferenceManager
import com.androidnetworking.AndroidNetworking
import com.androidnetworking.common.Priority
import com.androidnetworking.error.ANError
import com.androidnetworking.interfaces.JSONObjectRequestListener
import com.bumptech.glide.Glide
import org.json.JSONException
import org.json.JSONObject
```

```
28 import java.io.File
29 import java.net.URISyntaxException
30
31
32 class request : AppCompatActivity() {
33    lateinit var messageText: EditText
34     lateinit var chooseImageText: TextView
35     lateinit var postImage: ImageView
36     lateinit var submit_button: Button
37     var imageUri: Uri? = null
38     @RequiresApi(Build.VERSION_CODES.M)
39     override fun onCreate(savedInstanceState: Bundle?) {
40         super.onCreate(savedInstanceState)
41         setContentView(R.layout.activity_requestactivity)
42         AndroidNetworking.initialize(applicationContext)
43         messageText = findViewById(R.id.message)
44         chooseImageText = findViewById(R.id.choose_text)
45         postImage = findViewById(R.id.post_image)
46         submit_button = findViewById(R.id.submit_button)
47         submit_button.setOnClickListener(View.OnClickListener {
48             if (isValid) {
49                 //code to upload this post.
50                 uploadRequest(messageText.getText().toString())
51             }
52         })
53         chooseImageText.setOnClickListener(View.OnClickListener { //code to pick
   image
```

```
54              permission()
55          })
56      }
57
58      private fun pickImage() {
59          val intent = Intent(Intent.ACTION_GET_CONTENT)
60          intent.type = "image/*"
61          startActivityForResult(intent, 101)
62      }
63
64      @RequiresApi(Build.VERSION_CODES.M)
65      private fun permission() {
66          if (PermissionChecker.checkSelfPermission(
67                  applicationContext,
68                  permission.READ_EXTERNAL_STORAGE
69              )
70              != PermissionChecker.PERMISSION_GRANTED
71          ) {
72              //asking for permission
73              requestPermissions(arrayOf(permission.READ_EXTERNAL_STORAGE), 401)
74          } else {
75              //permission is already there
76              pickImage()
77          }
78      }
79
80      override fun onRequestPermissionsResult(
```

```
 81          requestCode: Int, permissions: Array<String>,
 82          grantResults: IntArray
 83      ) {
 84          super.onRequestPermissionsResult(requestCode, permissions, grantResults)
 85          if (requestCode == 401) {
 86              if (grantResults[0] == PermissionChecker.PERMISSION_GRANTED) {
 87                  //permission was granted
 88                  pickImage()
 89              } else {
 90                  //permission not granted
 91                  showMessage("Permission Declined")
 92              }
 93          }
 94      }
 95
 96      private fun uploadRequest(message: String) {
 97          //code to upload the message
 98          var path: String? = ""
 99          try {
100              path = getPath(imageUri)
101          } catch (e: URISyntaxException) {
102              showMessage("wrong uri")
103          }
104          val number =
105              PreferenceManager.getDefaultSharedPreferences(applicationContext)
106                  .getString("number", "12345")
107          AndroidNetworking.upload(upload_url)
```

```
108                 .addMultipartFile("file", File(path))
109                 .addQueryParameter("message", message)
110                 .addQueryParameter("number", number)
111                 .setPriority(Priority.HIGH)
112                 .build()
113                 .setUploadProgressListener { bytesUploaded, totalBytes -> // do
    anything with progress
114                     val progress = bytesUploaded / totalBytes * 100
115                     chooseImageText!!.text = "$progress%"
116                     chooseImageText!!.setOnClickListener(null)
117                 }
118                 .getAsJSONObject(object : JSONObjectRequestListener {
119                     override fun onResponse(response: JSONObject) {
120                         try {
121                             if (response.getBoolean("success")) {
122                                 showMessage("Succesfull")
123                                 finish()
124                             } else {
125                                 showMessage(response.getString("message"))
126                             }
127                         } catch (e: JSONException) {
128                             e.printStackTrace()
129                         }
130                     }
131
132                     override fun onError(anError: ANError) {}
133                 })
```

```
134     }
135
136     override fun onActivityResult(
137         requestCode: Int,
138         resultCode: Int,
139         data: Intent?
140     ) {
141         super.onActivityResult(requestCode, resultCode, data)
142         if (requestCode == 101 && resultCode == Activity.RESULT_OK) {
143             if (data != null) {
144                 imageUri = data.data
145                 Glide.with(applicationContext).load(imageUri).into(postImage!!)
146             }
147         }
148     }
149
150     private val isValid: Boolean
151         private get() {
152             if (messageText!!.text.toString().isEmpty()) {
153                 showMessage("Message shouldn't be empty")
154                 return false
155             } else if (imageUri == null) {
156                 showMessage("Pick Image")
157                 return false
158             }
159             return true
160         }
```

```
161
162     private fun showMessage(msg: String) {
163         Toast.makeText(this, msg, Toast.LENGTH_SHORT).show()
164     }
165
166     @SuppressLint("NewApi")
167     @Throws(URISyntaxException::class)
168     private fun getPath(uri: Uri?): String? {
169         var uri = uri
170         val needToCheckUri = Build.VERSION.SDK_INT >= 19
171         var selection: String? = null
172         var selectionArgs: Array<String>? = null
173         // Uri is different in versions after KITKAT (Android 4.4), we need to
174         // deal with different Uris.
175         if (needToCheckUri && DocumentsContract.isDocumentUri(applicationContext,
    uri)) {
176             if (isExternalStorageDocument(uri)) {
177                 val docId = DocumentsContract.getDocumentId(uri)
178                 val split = docId.split(":").toTypedArray()
179                 return Environment.getExternalStorageDirectory()
180                     .toString() + "/" + split[1]
181             } else if (isDownloadsDocument(uri)) {
182                 val id = DocumentsContract.getDocumentId(uri)
183                 uri = ContentUris.withAppendedId(
184                     Uri.parse("content://downloads/public_downloads"),
185                     java.lang.Long.valueOf(id)
186                 )
```

```
187             } else if (isMediaDocument(uri)) {
188                 val docId = DocumentsContract.getDocumentId(uri)
189                 val split = docId.split(":").toTypedArray()
190                 val type = split[0]
191                 if ("image" == type) {
192                     uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI
193                 } else if ("video" == type) {
194                     uri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI
195                 } else if ("audio" == type) {
196                     uri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI
197                 }
198                 selection = "_id=?"
199                 selectionArgs = arrayOf(
200                     split[1]
201                 )
202             }
203         }
204         if ("content".equals(uri!!.scheme, ignoreCase = true)) {
205             val projection = arrayOf(
206                 MediaStore.Images.Media.DATA
207             )
208             var cursor: Cursor? = null
209             try {
210                 cursor = contentResolver
211                     .query(uri, projection, selection, selectionArgs, null)
212                 val column_index = cursor!!.getColumnIndexOrThrow(MediaStore.Images
    .Media.DATA)
```

```
213                     if (cursor.moveToFirst()) {
214                         return cursor.getString(column_index)
215                     }
216                 } catch (e: Exception) {
217                 }
218             } else if ("file".equals(uri.scheme, ignoreCase = true)) {
219                 return uri.path
220             }
221             return null
222         }
223
224         companion object {
225             fun isExternalStorageDocument(uri: Uri?): Boolean {
226                 return "com.android.externalstorage.documents" == uri!!.authority
227             }
228
229             fun isDownloadsDocument(uri: Uri?): Boolean {
230                 return "com.android.providers.downloads.documents" == uri!!.authority
231             }
232
233             fun isMediaDocument(uri: Uri?): Boolean {
234                 return "com.android.providers.media.documents" == uri!!.authority
235             }
236         }
237 }
```

```kotlin
package com.example.myapplication

import VolleySingleton
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.Toolbar
import androidx.preference.PreferenceManager
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.android.volley.AuthFailureError
import com.android.volley.Response
import com.android.volley.toolbox.StringRequest
import com.google.gson.Gson
import com.google.gson.reflect.TypeToken
import java.util.*
import kotlin.collections.ArrayList

class MainActivity : AppCompatActivity() {


    lateinit var recyclerview : RecyclerView
    lateinit var Adapter : RequestAdapter
    private var RequestDataModels: MutableList<RequestDataModel> = ArrayList()
```

```
28      lateinit var btn : TextView
29
30      override fun onCreate(savedInstanceState: Bundle?) {
31          super.onCreate(savedInstanceState)
32          setContentView(R.layout.activity_main)
33
34          btn = findViewById<TextView>(R.id.make_request_button)
35
36          btn.setOnClickListener {
37              startActivity(Intent(this , request::class.java))
38
39          }
40
41
42          val toolbar = findViewById<Toolbar>(R.id.toolbar)
43          toolbar.setOnMenuItemClickListener { item ->
44              if (item.itemId == R.id.search_button) {
45                  //open search
46                  startActivity(Intent(this, SearchActivity::class.java))
47              }
48              true
49          }
50
51
52          recyclerview = findViewById(R.id.recyclerView)
53          recyclerview.layoutManager = LinearLayoutManager(this, RecyclerView.VERTICAL
    , false)
```

```
54          Adapter  = RequestAdapter(RequestDataModels, this)
55          recyclerview.setAdapter(Adapter)
56          populateHomePage()
57
58
59
60      }
61      private fun populateHomePage() {
62          val city =
63              PreferenceManager.getDefaultSharedPreferences(applicationContext)
64                  .getString("city", "no_city")
65          val stringRequest: StringRequest = object : StringRequest(
66              Method.POST,
67              get_url,
68              Response.Listener { response ->
69                  val gson = Gson()
70                  val type = object :
71                      TypeToken<List<RequestDataModel?>?>() {}.type
72                  val dataModels =
73                      gson.fromJson<List<RequestDataModel>>(response, type)
74                  RequestDataModels!!.addAll(dataModels)
75                  Adapter!!.notifyDataSetChanged()
76              },
77              Response.ErrorListener { error ->
78                  Toast.makeText(this@MainActivity, "Something went wrong:(", Toast.
   LENGTH_SHORT)
79                      .show()
```

```
80                  Log.d(
81                      "VOLLEY",
82                      Objects.requireNonNull(error.message)
83                  )
84              }) {
85              @Throws(AuthFailureError::class)
86              override fun getParams(): Map<String, String> {
87                  val params: MutableMap<String, String> =
88                      HashMap()
89
90                  return params
91              }
92          }
93          VolleySingleton.getInstance(this)!!.addToRequestQueue(stringRequest)
94      }
95  }
96
```

File - C:\Users\C GIRI TEJA\Downloads\Blood-Bank-master\Blood-Bank-master\app\src\main\java\com\example\myapplication\splashscreen.kt

```kotlin
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import androidx.appcompat.app.AppCompatActivity

class splashscreen : AppCompatActivity() {

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









        Handler().postDelayed({
            /* Create an Intent that will start the Menu-Activity. */
            val mainIntent = Intent(this, loginactivity::class.java)
            startActivity(mainIntent)
            finish()
        }, 1000)

    }
```

```
28 }
29
```

```kotlin
package com.example.myapplication

import VolleySingleton
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.preference.PreferenceManager
import com.android.volley.AuthFailureError
import com.android.volley.Request
import com.android.volley.Response
import com.android.volley.toolbox.StringRequest


class loginactivity : AppCompatActivity() {
lateinit var numberET : EditText
lateinit var passwordET :EditText
lateinit var submit_button : Button
lateinit var signupText: TextView
lateinit var tv : TextView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
28          setContentView(R.layout.activity_loginactivity)
29          numberET = findViewById<EditText>(R.id.username)
30          passwordET = findViewById<EditText>(R.id.password)
31          submit_button = findViewById<Button>(R.id.submit_button)
32          signupText =  findViewById<TextView>(R.id.sign_up_text)
33          tv =findViewById<TextView>(R.id.tv)
34          signupText.setOnClickListener(View.OnClickListener {
35              startActivity(Intent(this , registeractivity::class.java))
36
37          })
38          submit_button.setOnClickListener(View.OnClickListener {
39                  numberET.setError(null)
40                  passwordET.setError(null)
41                  var number :String = numberET.text.toString()
42                  var passworrd : String = passwordET.text.toString()
43                  if(isValid(number,passworrd))
44                  {
45                      login(number,passworrd)
46                  }
47          })
48          tv.setOnClickListener(View.OnClickListener {
49
50          })
51
52      }
53      private fun login(number : String ,password :String){
54          val stringrequest : StringRequest = object : StringRequest(Request.Method.
```

```
54 POST,
55             login_url,Response.Listener<String>{ response->
56             if(response == "success"){
57                 Toast.makeText(this,response , Toast.LENGTH_SHORT).show()
58                 startActivity(Intent(this,MainActivity::class.java))
59                 PreferenceManager.getDefaultSharedPreferences(applicationContext).
   edit().putString("number",number).apply()
60                 finish()
61             }else{
62                 Toast.makeText(this,response,Toast.LENGTH_SHORT).show()
63             }
64
65
66         },Response.ErrorListener{error ->
67             Toast.makeText(this,"Something went wrong",Toast.LENGTH_SHORT).show()
68             Log.d("Volley",error.message)
69
70         })
71         {
72             @Throws(AuthFailureError::class)
73             override fun getParams(): MutableMap<String, String> {
74                 val params : MutableMap<String,String> =HashMap()
75
76                 params["number"]=number
77                 params["password"]=password
78
79
```

```
80                  return params
81
82
83              }
84          }
85          VolleySingleton.getInstance(this)!!.addToRequestQueue(stringrequest)
86
87      }
88
89      fun show(str : String)
90      {
91          Toast.makeText(this,str,Toast.LENGTH_SHORT).show()
92      }
93
94      private fun isValid(number :String , passworrd: String) : Boolean{
95          if(number.isEmpty())
96          {
97              show("Number column is empty")
98              numberET.setError("Empty number")
99              return false
100         }else if(passworrd.isEmpty()){
101             show("password column is empty")
102             numberET.setError("Empty password")
103             return false
104         }
105
106         return true
```

```
107     }
108
109 }
110
```

```kotlin
package com.example.myapplication

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import kotlinx.android.synthetic.main.request_layout.view.*


class RequestAdapter(
    private val dataSet: MutableList<RequestDataModel>, private val context: Context
) : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
    override fun onCreateViewHolder(
        parent: ViewGroup,
        viewType: Int
    ): ViewHolder {
        val view: View = LayoutInflater.from(parent.context)
            .inflate(R.layout.request_layout, parent, false)
        return ViewHolder(view)
    }
    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {


```

```
28          holder.itemView.message.text = dataSet.get(position).message
29          Glide.with(context).load(dataSet[position].url).into(holder.itemView.image)
30          holder.itemView.callbutton.setOnClickListener{}
31
32
33          holder.itemView.sharebutton.setOnClickListener() {
34          }
35      }
36
37
38      override fun getItemCount(): Int {
39          return dataSet.size
40      }
41
42      inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
43          lateinit var message: TextView
44          lateinit var imageView: ImageView
45          lateinit var callButton: ImageView
46          lateinit var shareButton: ImageView
47         init{
48
49              message = itemView.findViewById(R.id.message)
50              imageView = itemView.findViewById(R.id.image)
51              callButton = itemView.findViewById(R.id.callbutton)
52              shareButton = itemView.findViewById(R.id.sharebutton)
53
54          }
```

```
55
56      }
57
58
59
60
61 }
```

```kotlin
package com.example.myapplication

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class SearchActivity : AppCompatActivity() {

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

```

```kotlin
import android.content.Context
import com.android.volley.Request
import com.android.volley.RequestQueue
import com.android.volley.toolbox.Volley

class VolleySingleton private constructor(private val mContext: Context) {
    private var mRequestQueue: RequestQueue?
    // If RequestQueue is null the initialize new RequestQueue

    // Return RequestQueue
    val requestQueue: RequestQueue?
        get() {
            // If RequestQueue is null the initialize new RequestQueue
            if (mRequestQueue == null) {
                mRequestQueue =
                    Volley.newRequestQueue(mContext.applicationContext)
            }

            // Return RequestQueue
            return mRequestQueue
        }

    fun <T> addToRequestQueue(request: Request<T>?) {
        // Add the specified request to the request queue
        requestQueue!!.add(request)
    }

```

```
28      companion object {
29          private var mInstance: VolleySingleton? = null
30
31          @Synchronized
32          fun getInstance(context: Context): VolleySingleton? {
33              // If Instance is null then initialize new Instance
34              if (mInstance == null) {
35                  mInstance = VolleySingleton(context)
36              }
37              // Return MySingleton new Instance
38              return mInstance
39          }
40      }
41
42      init {
43          // Specify the application context
44          // Get the request queue
45          mRequestQueue = requestQueue
46      }
47  }
```

```kotlin
package com.example.myapplication

import VolleySingleton
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.android.volley.AuthFailureError
import com.android.volley.Response
import com.android.volley.toolbox.StringRequest as StringRequest1



class registeractivity : AppCompatActivity() {
    lateinit var nameET: EditText

    lateinit var cityET: EditText
    lateinit var passwordET: EditText
    lateinit var numberET: EditText
    lateinit var bloodgroupET: EditText
    lateinit var submit: Button
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
28          setContentView(R.layout.activity_registeractivity)
29
30          nameET = findViewById<EditText>(R.id.name)
31          cityET = findViewById<EditText>(R.id.city)
32          passwordET = findViewById<EditText>(R.id.password)
33          numberET = findViewById<EditText>(R.id.number)
34          bloodgroupET = findViewById<EditText>(R.id.bloodgroup)
35          submit = findViewById<Button>(R.id.submitbutton)
36          submit.setOnClickListener(View.OnClickListener {
37              var name: String = nameET.text.toString()
38              var city: String = cityET.text.toString()
39              var password: String = passwordET.text.toString()
40              var number: String = numberET.text.toString()
41              var bloodgroup: String = bloodgroupET.text.toString()
42              show(name + "\n" + city + "\n" + password + "\n" + number + "\n" +
    bloodgroup)
43              if(isValid(name, city, password, number, bloodgroup)) {
44                  register(name, city, bloodgroup, password, number)
45              }
46
47          })
48      }
49      private fun register(name:String, city:String, blood_group: String, password:
    String, number: String){
50      val stringRequest: StringRequest1 = object : StringRequest1(
51          Method.POST, register_url, Response.Listener { response ->
52              if (response == "success") {
```

```
53
54                  Toast.makeText(this,response, Toast.LENGTH_SHORT).show()
55                  val mainIntent = Intent(this, MainActivity::class.java)
56                  startActivity(mainIntent)
57                  finish()
58
59              } else {
60                  Toast.makeText(this, response, Toast.LENGTH_SHORT).show()
61              }
62          },
63          Response.ErrorListener { error ->
64              Toast.makeText(
65                  this, "Something went wrong:(", Toast.LENGTH_SHORT).show()
66              Log.d("VOLLEY", error.message)
67          }) {
68          @Throws(AuthFailureError::class)
69          override fun getParams(): Map<String, String> {
70              val params: MutableMap<String, String> =
71                  HashMap()
72
73              params["name"] = name
74              params["city"] = city
75              params["blood_groups"] = blood_group
76              params["password"] = password
77              params["number"] = number
78              return params
79          }
```

```
 80      }
 81      VolleySingleton.getInstance(this)!!.addToRequestQueue(stringRequest)
 82  }
 83
 84  fun isValid(
 85      name: String,
 86      city: String,
 87      password: String,
 88      number: String,
 89      bloodgroup: String
 90  ): Boolean {
 91      var valid_list: ArrayList<String> = ArrayList()
 92      valid_list.add("A+")
 93      valid_list.add("A-")
 94      valid_list.add("AB+")
 95      valid_list.add("AB-")
 96      valid_list.add("O+")
 97      valid_list.add("O-")
 98      valid_list.add("B+")
 99      valid_list.add("B-")
100
101      if (name.isEmpty()) {
102          show("Name is empty")
103          return false;
104      } else if (city.isEmpty()) {
105          show("City is empty")
106          return false;
```

```
107     } else if (!valid_list.contains(bloodgroup)) {
108         show("Invalid Blood Group$bloodgroup")
109         return false
110     } else if (number.length != 10) {
111         show("invalid mobile number ,must be of 10 digits")
112         return false
113     }
114
115
116
117     return true;
118
119 }
120
121 fun show(str: String) {
122     Toast.makeText(this, str, Toast.LENGTH_SHORT).show()
123 }
124 }
125
```

```kotlin
package com.example.myapplication

import com.google.gson.annotations.Expose
import com.google.gson.annotations.SerializedName


class RequestDataModel {
    @SerializedName("id")
    @Expose
    var id: String? = null

    @SerializedName("message")
    @Expose
    var message: String? = null

    @SerializedName("url")
    @Expose
    var url: String? = null

    @SerializedName("number")
    @Expose
    var number: String? = null

}


```