manifest:

```
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
   <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

AboutActivity

```
package com.example.model
import android.os.Bundle
import android.view.ContextMenu
import android.view.MenuItem
import android.view.View
import android.widget.Button
import android.widget.PopupMenu
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class AboutActivity : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.about)
      val btn = findViewById<Button>(R.id.contact)
      btn.setOnClickListener{
            view -> showPopUp(view)
      }
      val rbtn = findViewById<Button>(R.id.rate)
      registerForContextMenu(rbtn)
   }
   private fun showPopUp(view: View?) {
      val popup = PopupMenu(this, view)
      popup.menuInflater.inflate(R.menu.popup_menu, popup.menu)
      popup.setOnMenuItemClickListener { item ->
         when (item.itemId) {
            R.id.contactphn -> {
               Toast.makeText(this, "Call this no 9751423916",
Toast.LENGTH_SHORT).show()
               true
            }
            R.id.contactMail -> {
               Toast.makeText(this, "Mail sakthisarvani@studen.tce.edu",
```

```
Toast.LENGTH_SHORT).show()
                true
            }
            else -> false
        }
    }
    popup.show()
  }
  override fun onCreateContextMenu(
    menu: ContextMenu?,
    v: View?,
    menuInfo: ContextMenu.ContextMenuInfo?
  ) {
    super.onCreateContextMenu(menu, v, menuInfo)
    menuInflater.inflate(R.menu.context_menu,menu)
  }
  override fun onContextItemSelected(item: MenuItem): Boolean {
    return when(item.itemId){
      R.id.r1->{
        Toast.makeText(this, "Thanks for rating 1 star",
Toast.LENGTH_SHORT).show()
        true
      }
      R.id.r2->{
        Toast.makeText(this, "Thanks for rating 2 star",
Toast.LENGTH_SHORT).show()
        true
      }
      else-> super.onContextItemSelected(item)
    }
  }
}
```

BookingActivity:

```
package com.example.model

import android.app.NotificationChannel
import android.app.NotificationManager
import android.content.Context
import android.os.Build
```

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.TextView
import androidx.core.app.NotificationCompat
class BookingActivity : AppCompatActivity() {
    private val CHANNEL_ID = "booking_channel"
    private val NOTIFICATION_ID = 101
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_booking)
        val bookingInfo = findViewById<TextView>(R.id.bookingInfo)
        val eventDetails = intent.getStringExtra("eventDetails") ?: "Unknown Event"
        bookingInfo.text = "Booking Confirmed!\nEvent: $eventDetails\nTime: 7:00 PM"
        createNotificationChannel()
        sendBookingNotification(eventDetails)
    }
    private fun createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val name = "Booking Notifications"
            val descriptionText = "Notifications for event bookings"
            val importance = NotificationManager.IMPORTANCE_DEFAULT
            val channel = NotificationChannel(CHANNEL_ID, name, importance).apply {
                description = descriptionText
            }
            val notificationManager: NotificationManager =
                getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
            notificationManager.createNotificationChannel(channel)
        }
    }
    private fun sendBookingNotification(eventDetails: String) {
        val notificationBuilder = NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(android.R.drawable.ic_dialog_info)
            .setContentTitle("Booking Confirmation")
            .setContentText("Your booking for $eventDetails has been confirmed")
            .setStyle(NotificationCompat.BigTextStyle()
                .bigText("Your booking for $eventDetails has been confirmed. The event starts
at 7:00 PM. Don't forget to arrive on time!"))
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)
            .setAutoCancel(true)
        val notificationManager = getSystemService(Context.NOTIFICATION_SERVICE)
```

```
as NotificationManager
        notificationManager.notify(NOTIFICATION_ID, notificationBuilder.build())
    }
}
```

HomeActivity:

```
package com.example.model

import android.app.AlertDialog
import android.content.Intent
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.ArrayAdapter
import android.widget.Button
import android.widget.GridView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class HomeActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.homeacivity)
        val events = arrayOf(
            "Concert - April 10",
            "Movie Night - April 12",
            "Art Gallery - April 15",
            "Food Fest - April 18",
            "Tech Talk - April 20",
            "Sports Day - April 22"
        )
        val gridView = findViewById<GridView>(R.id.gridView)
        val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, events)
        gridView.adapter = adapter
        gridView.setOnItemClickListener { _, _, position, _ ->
            val selectedEvent = events[position]
            val builder = AlertDialog.Builder(this)
            builder.setTitle("Book Now")
            builder.setMessage("Book $selectedEvent?")
            builder.setPositiveButton("Yes") { _, _ ->
                val intent = Intent(this, BookingActivity::class.java).apply {
                    putExtra("eventDetails", selectedEvent)
```

```
            }
            startActivity(intent)
        }
        builder.setNegativeButton("No", null)
        builder.show()
    }
    val exitBtn = findViewById<Button>(R.id.exitbtn)
    exitBtn.setOnClickListener {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Exit Application")
        builder.setMessage("Are you sure want to exit?")
        builder.setPositiveButton("Yes") { _, _ -> finishAffinity() }
        builder.setNegativeButton("No", null)
        builder.show()
    }
}
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.options_menu, menu)
    return true
}
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.aboutus -> {
            val intent = Intent(this, AboutActivity::class.java)
            startActivity(intent)
            true
        }
        R.id.profile -> {
            val intent = Intent(this, ProfileActivity::class.java)
            startActivity(intent)
            true
        }
        R.id.settings -> {
            val intent = Intent(this, SettingsActivity::class.java)
            startActivity(intent)
            Toast.makeText(this, "Settings menu selected",
Toast.LENGTH_SHORT).show()
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
```

```
    }
}
```

LoginActivity:

```
package com.example.model
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.login)
        val sharedPref = getSharedPreferences("userPrefs", Context.MODE_PRIVATE)
        val btn = findViewById<Button>(R.id.login)
        btn.setOnClickListener {
            val uname = findViewById<EditText>(R.id.editTextText2).text.toString()
            val email =
findViewById<EditText>(R.id.editTextTextEmailAddress2).text.toString()
            val editor = sharedPref.edit()
            editor.putString("username",uname)
            editor.putString("email",email)
            editor.apply()
            val intent = Intent(this, HomeActivity::class.java)
            Toast.makeText(this, "Explicit Intent is activated",
Toast.LENGTH_SHORT).show()
            startActivity(intent)
        }
    }
}
```

MainActivity:

```
package com.example.model

import android.annotation.SuppressLint
import android.app.AlertDialog
import android.content.Intent
```

```
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.Button
import android.widget.LinearLayout
import android.widget.ProgressBar
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
class MainActivity : AppCompatActivity() {
    @SuppressLint("MissingInflatedId")
    private lateinit var pbar:ProgressBar
    private lateinit var main:LinearLayout
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        main=findViewById(R.id.mainLinear)
        main.visibility= View.GONE
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v,
insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }
        pbar = findViewById<ProgressBar>(R.id.bar)
        Thread{
            Thread.sleep(10)
            for(i in 1..10){
                runOnUiThread{
                    pbar.progress = i
                }
            }
            runOnUiThread{
                main.visibility=View.VISIBLE
            }
        }.start()
```

```
        val btn = findViewById<Button>(R.id.sign)
        btn.setOnClickListener {
            val intent = Intent(this, SignUpActivity::class.java)
            Toast.makeText(this, "Explicit Intent is activated",
Toast.LENGTH_SHORT).show()
            startActivity(intent)
        }
        val btn1 = findViewById<Button>(R.id.log)
        btn1.setOnClickListener {
            val intent = Intent(this, LoginActivity::class.java)
            Toast.makeText(this, "Explicit Intent is activated",
Toast.LENGTH_SHORT).show()
            startActivity(intent)
        }
    }
    override fun onBackPressed() {
        super.onBackPressed()
        val builder = AlertDialog.Builder(this)
        with(builder) {
            setTitle("Exit Application")
            setMessage("Are you sure want to exit the application")
            setPositiveButton("Yes") { _, _ -> finishAffinity() }
            setNegativeButton("NO", null)
            setCancelable(true)
            show()
        }
    }
}
```

ProfileActivity:

```
package com.example.model
import android.annotation.SuppressLint
import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
class ProfileActivity : AppCompatActivity() {
    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.profile)
        val sharedPref = getSharedPreferences("userPrefs",Context.MODE_PRIVATE)
        val name = findViewById<TextView>(R.id.uname)
        val mail = findViewById<TextView>(R.id.email)
        val uname = sharedPref.getString("username","Guest")
        val email = sharedPref.getString("email","Not found")
        name.text = "Username : $uname"
        mail.text = "Email : $email"
    }
}
```

SettingsActivity:

```
package com.example.model
import android.app.DatePickerDialog
import android.app.TimePickerDialog
import android.location.Geocoder
import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import java.util.*
import android.Manifest
import android.content.pm.PackageManager
import android.location.Location
import androidx.core.app.ActivityCompat
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
class SettingsActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.settings)
        val cal = Calendar.getInstance()
        val dateBtn = findViewById<Button>(R.id.dateBtn)
        val timeBtn = findViewById<Button>(R.id.timeBtn)
        val dateText = findViewById<TextView>(R.id.dateText)
        val timeText = findViewById<TextView>(R.id.timeText)
        val latInput = findViewById<EditText>(R.id.latInput)
        val lonInput = findViewById<EditText>(R.id.lonInput)
        val fetchBtn = findViewById<Button>(R.id.fetchBtn)
        val resultText = findViewById<TextView>(R.id.resultText)
        val progressBar = findViewById<ProgressBar>(R.id.progressBar)
        val currentLocationBtn = findViewById<Button>(R.id.currentLocationBtn)
```

```
        val currentLocationText = findViewById<TextView>(R.id.currentLocationText)
        val fusedLocationClient: FusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this)
        currentLocationBtn.setOnClickListener {
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), 101)
                return@setOnClickListener
            }
            fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->
                if (location != null) {
                    val lat = location.latitude
                    val lon = location.longitude
                    val geocoder = Geocoder(this, Locale.getDefault())
                    val addresses = geocoder.getFromLocation(lat, lon, 1)
                    if (!addresses.isNullOrEmpty()) {
                        val address = addresses[0].getAddressLine(0)
                        currentLocationText.text = "Current Location:\nLat: $lat\nLon:
$lon\n$address"
                    } else {
                        currentLocationText.text = "Location found, but address not available."
                    }
                } else {
                    currentLocationText.text = "Couldn't get location. Try again."
                }
            }
        }
        dateBtn.setOnClickListener {
            val year = cal.get(Calendar.YEAR)
            val month = cal.get(Calendar.MONTH)
            val day = cal.get(Calendar.DAY_OF_MONTH)
            DatePickerDialog(this, { _, y, m, d ->
                dateText.text = "Date: $d/${m + 1}/$y"
            }, year, month, day).show()
        }
        timeBtn.setOnClickListener {
            val hour = cal.get(Calendar.HOUR_OF_DAY)
            val min = cal.get(Calendar.MINUTE)
            TimePickerDialog(this, { _, h, m ->
```

```
            timeText.text = "Time: $h:$m"
        }, hour, min, true).show()
    }
    fetchBtn.setOnClickListener {
        val lat = latInput.text.toString().toDoubleOrNull()
        val lon = lonInput.text.toString().toDoubleOrNull()
        if (lat == null || lon == null) {
            Toast.makeText(this, "Enter valid lat/lon", Toast.LENGTH_SHORT).show()
            return@setOnClickListener
        }
        progressBar.progress = 0
        resultText.text = ""
        Thread {
            for (i in 1..100) {
                Thread.sleep(30)
                runOnUiThread {
                    progressBar.progress = i
                }
            }
            val geocoder = Geocoder(this, Locale.getDefault())
            try {
                val addresses = geocoder.getFromLocation(lat, lon, 1)
                runOnUiThread {
                    if (addresses != null && addresses.isNotEmpty()) {
                        val address = addresses[0].getAddressLine(0)
                        resultText.text = "Address:\n$address"
                    } else {
                        resultText.text = "No address found!"
                    }
                }
            } catch (e: Exception) {
                runOnUiThread {
                    resultText.text = "Geocoder error: ${e.message}"
                }
            }
        }.start()
    }
  }
}
```

SignUpActivity:

```
package com.example.model
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class SignUpActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.signup)
        val btn = findViewById<Button>(R.id.signup)
        btn.setOnClickListener {
            val intent = Intent(this, HomeActivity::class.java)
            Toast.makeText(this, "Explicit Intent is activated",
Toast.LENGTH_SHORT).show()
            startActivity(intent)
        }
    }
}
```