**SmartInternLongTermVirtualInternship**

**An Internship Report Submitted In Partial Fulfillment of The Requirements for the Award of Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**MECH – MECHANICAL ENGINEERING**

**Submittedby:**

**RITHU PRADHAN 20W61A0323**

**SIMMA HEMANTH 20W61A0326**

**RATNALA RAKESH 21W65A0319**



**Department of Mechanical Engineering**

**SRI SIVANI COLLEGE OF ENGINEERING**

**Approved By AICTE & Affiliated to JNTUGV**

**Chilakapalem Jn., Srikakulam 532402**

SMARTINTERZVIRTUALINTERSHIP

**Part1:ExecutiveSummary**

Executive Summary: Altoro Mutual Website Vulnerability Exploitation and Patching

Altoro Mutual is a subsidiary of Altoro, a multi-state holding company located in the heart of

Massachusetts. Altoro Mutual has been serving Boston and surrounding communities for

nearly 75 years.

Altoro Mutual offers a broad range of commercial, private, retail and mortgage banking

services to small- and middle-market businesses and individuals.

Using a combination of manual analysis and automated scanning tools, the

assessmentaimed to detect vulnerabilities that could be exploited by malicious

actors to gainunauthorized access or compromise the website's sensitive data.

Rigorous testing wasperformed, taking into account various attack vectors and

techniques commonly used byhackers.

The assessment revealed several findings regarding the website's network

security.Multiple high-severity vulnerabilities were detected, including

unpatched software, openports with inadequate security controls, and

weaknesses in the password policyimplementation. These critical issues

exposed the website to potential cyberattacks,

databreaches,andservicedisruptions.

Toaddresstheidentifiedvulnerabilities,asetofcomprehensiverecommendationshas

been provided. Altoro Mutual can significantly enhance the security of its

website's

**Overview**

**Overview:Vulnerability Exploitation and Patching**

Vulnerability Exploitation and Patching on Altoro Mutual is a comprehensive evaluation

ofthe organization's network infrastructure to identify potential security weaknesses

andvulnerabilities. Altoro Mutual is a financial services company that handles sensitive

data,making it imperative to maintain a robust and secure network environment.

Thisassessment aims to identify and address security gaps that could expose the company

tocyberthreats,databreaches,andfinanciallosses.

The primary objectives of the Network Vulnerability Assessment on Altoro Mutual are

asfollows:

1. **Identify Vulnerabilities**: The assessment aims to identify potential vulnerabilities in

thenetworkinfrastructure,includingunpatchedsoftware,misconfigurations,andopenports.

2. **Evaluate Security Controls**: The effectiveness of existing security controls, such

asfirewalls,intrusiondetectionsystems(IDS),andaccesscontrols,isassessedtodeterminetheir

abilitytodetectandpreventattacks.

3. **AssessNetworkArchitecture:**Thenetworkarchitectureisreviewedtoensurepropersegme

ntation,isolationofcriticalassets,andarobustperimeterdefense.

4. **Password Policy Evaluation**: The assessment examines the strength of

passwordpoliciesandtheir adherencetoindustry bestpracticesto

preventunauthorizedaccess.

5. **Physical Security Analysis**: Physical security measures in place to protect

networkinfrastructureanddatacentersareevaluatedtopreventunauthorizedphysicalaccess

Methodology:

Theassessmentfollowsawell-definedmethodology,includingthefollowingsteps

1. Reconnaissance:Passivereconnaissancetechniquesareusedtogatherinformationabou

tthenetworkanditsassets.

2. Vulnerability Scanning: Automated scanning tools are employed to identify

potentialvulnerabilitiesinthenetwork.

3. Manual Verification: The identified vulnerabilities are manually verified to

eliminatefalsepositivesandprioritizecriticalissues.

4. Exploitation (with Authorization): Ethical exploitation of vulnerabilities is conducted

todeterminetheextentofpotentialdamageifexploitedmaliciously.

5. AnalysisandReporting:Theassessmentfindingsareanalyzed,andadetailedreportisgenerat

ed,includingalistofvulnerabilities,riskseverity,andactionablerecommendations.

Deliverables:

Theassessmentwillprovidethefollowingdeliverables:

1. Network Vulnerability Assessment Report: A comprehensive report detailing

theassessmentmethodology,findings,riskanalysis,andactionablerecommendations.

2. ExecutiveSummary:Aconcisesummaryhighlightingkeyfindingsandcriticalvulne

rabilitiesforexecutivestakeholders.

3. Remediation Plan: A roadmap outlining the prioritized actions required to

Addressidentifiedvulnerabilitiesandimprovenetworksecurity

### A Sleep Tracking App For A Better Night's Rest

### Project Description:

A Project That Demonstrates The Use Of Android Jetpack Compose To Build A UI For A Sleep Tracking App. The App Allows Users To Track Their Sleep.With The “Sleep Tracker” App, You Can Assess The Quality Of Sleep They Have Had In A Day. It Has Been Time And Again Proven That A Good Quality Sleep Is Pretty Essential For Effective Functioning Of Both Mind And Body.

### 

“Sleep Tracker” Application Enables You To Start The Timer When They Are In The Bed And About To Fall Asleep. The Timer Will Keep Running In The Background Until It Is Stopped, Whenever The User Wakes Up. Based On The Sleep Experience, You Can Rate Your Sleep Quality. Finally , The App Will Display An Analysis Of The Kind Of Sleep , You Had The Previous Night.

### 

### Architecture

### 

**Learning Outcomes :**

### 

By End Of This Project:

* You’ll Be Able To Work On Android Studio And Build An App.
* You’ll Be Able To Integrate The Database Accordingly.

### 

**Project Workflow:**

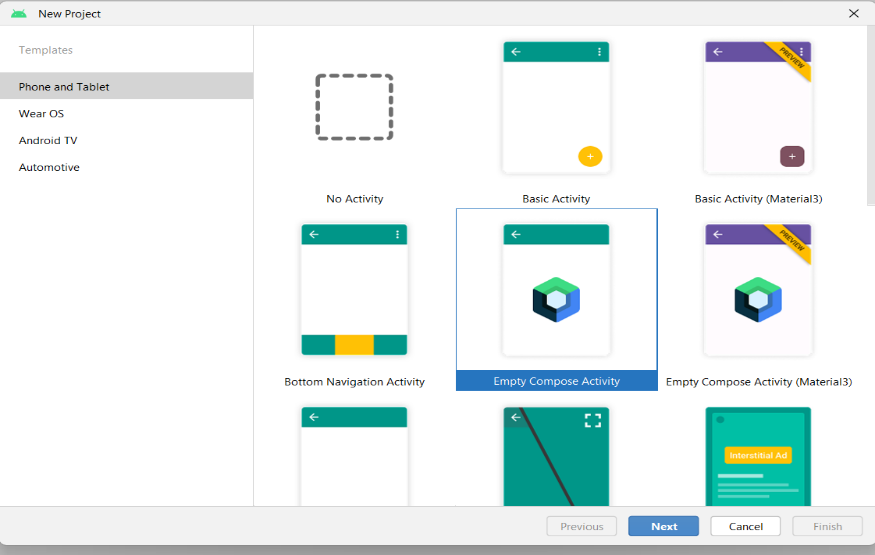
* Users Register Into The Application.
* After Registration , User Logins Into The Application.
* User Enters Into The Main Page
* User Can Track The Sleep Timing And He Record The Time

### Creating A New Project

**Creating a new project.**

Step 1 : Android studio > File > New > New Project > Empty Compose Activity

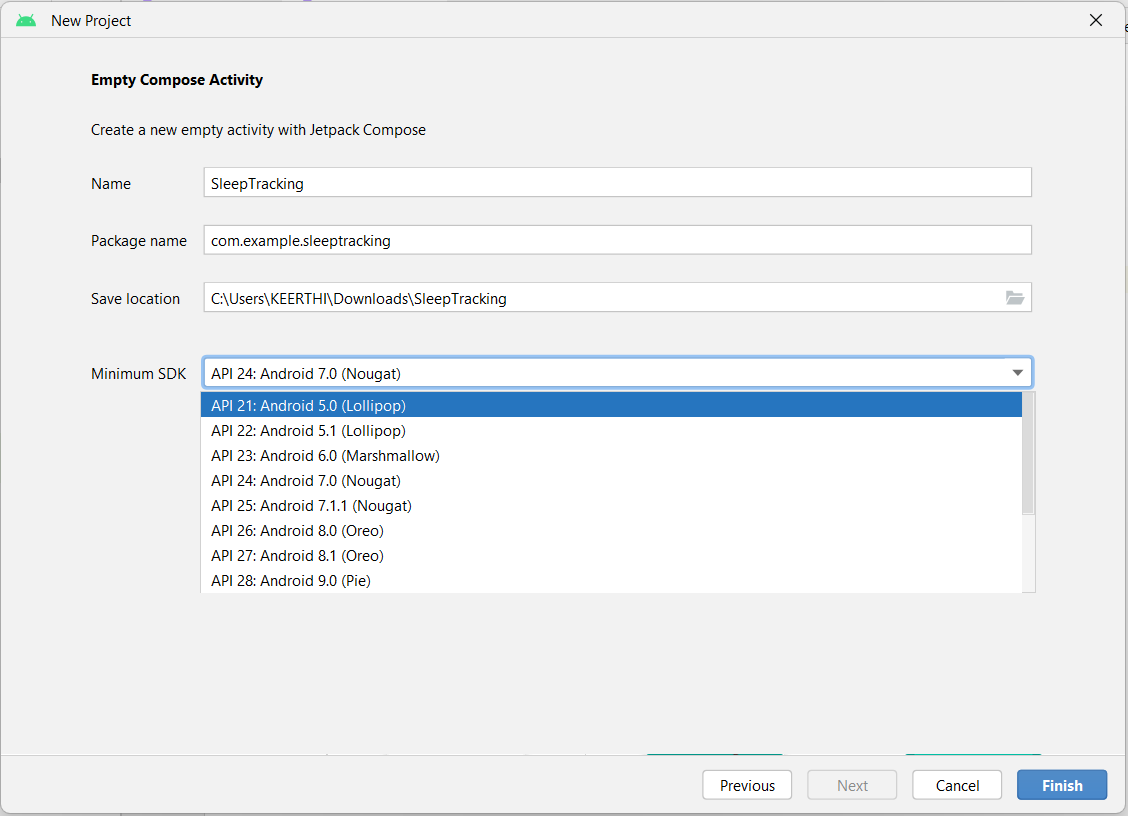
Step 2 : Click on **Next** button.



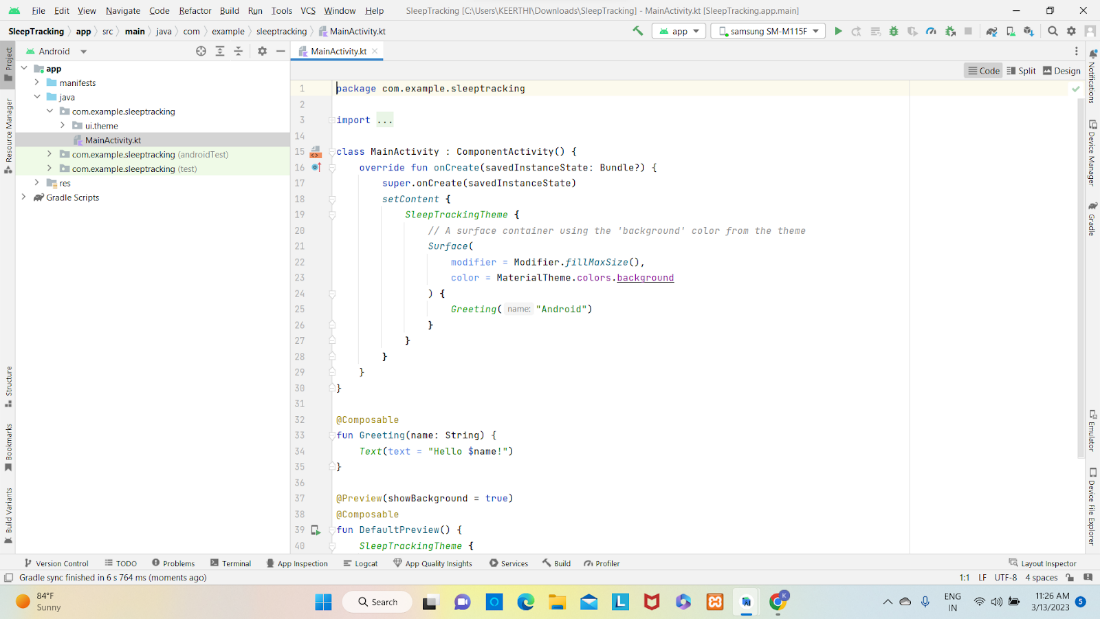
Step 3 : Give name to the new project.

Step 4 : Give the Minimum SDK value

Step 5 : Click Finish

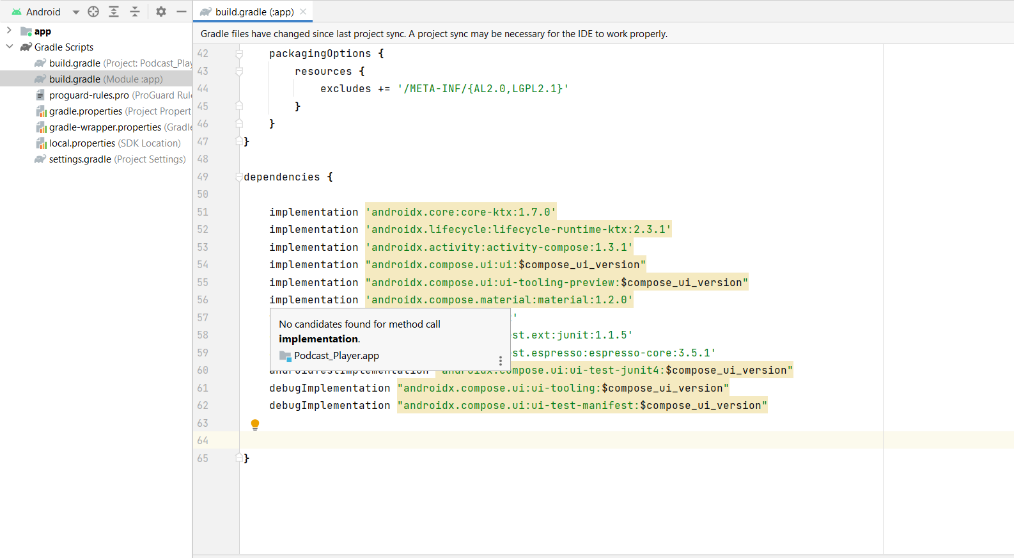


Main activity file



### Gradle Scripts > Build.Gradle(Module :App)

**Gradle scripts > build.gradle(Module :app)**



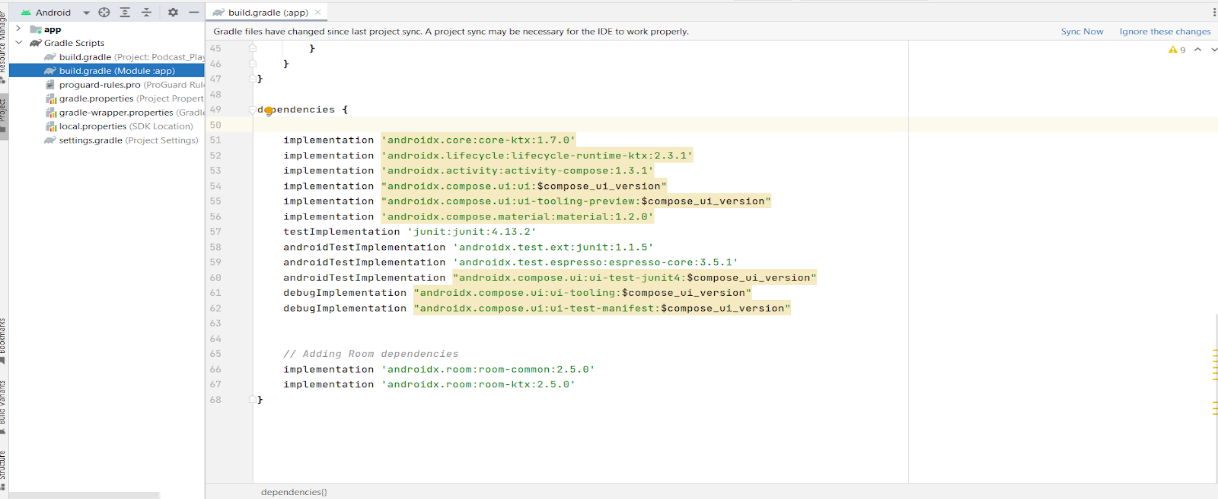
### Add The Below Code In Dependencies

**Add the below code in dependencies**

*// Adding Room dependencies*

implementation 'androidx.room:room-common:2.5.0'

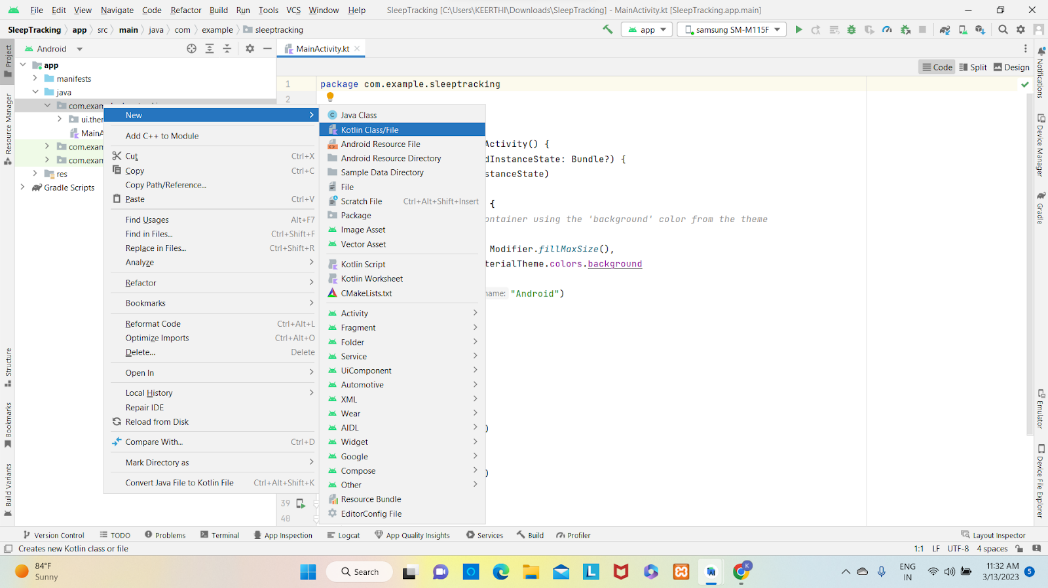
implementation 'androidx.room:room-ktx:2.5.0'

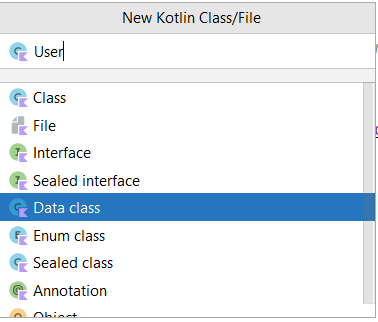


Step 3 : Click on Sync now

### Database 1 (Create User Data Class)

**(Create User data class)**



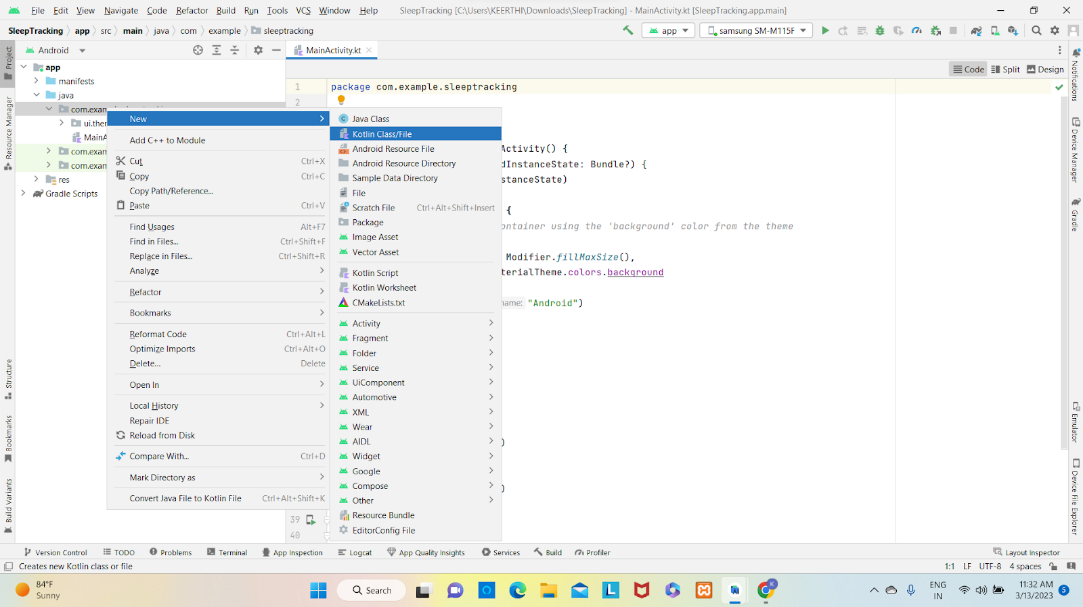


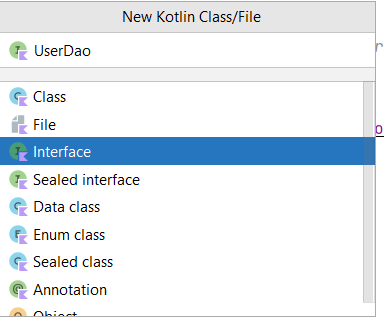
User class code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/User.kt>

### Create An UserDao Interface

**Create an UserDao interface**



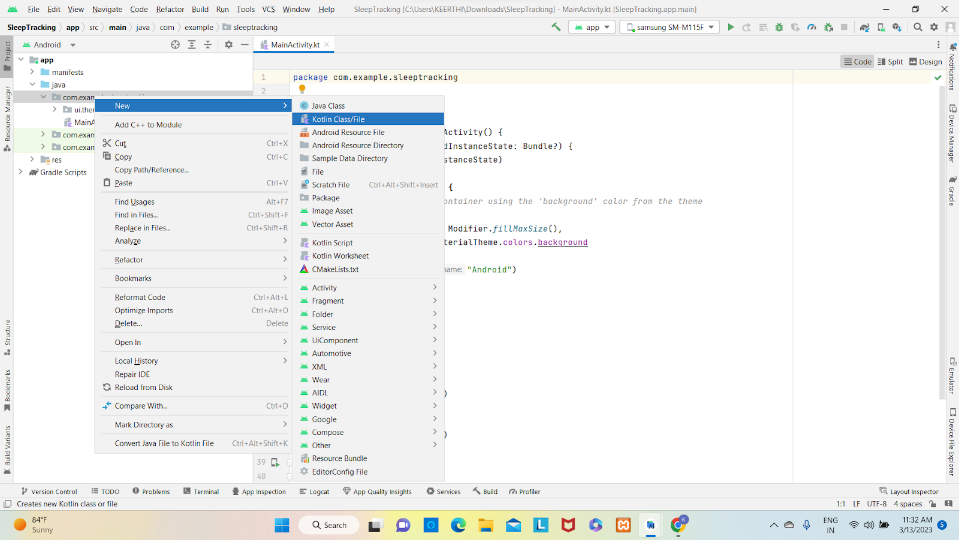


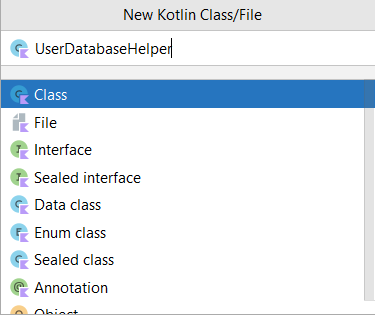
UserDao interface code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/UserDao.kt>

### Create An UserDatabaseHelper Class

**Create an UserDatabaseHelper class**



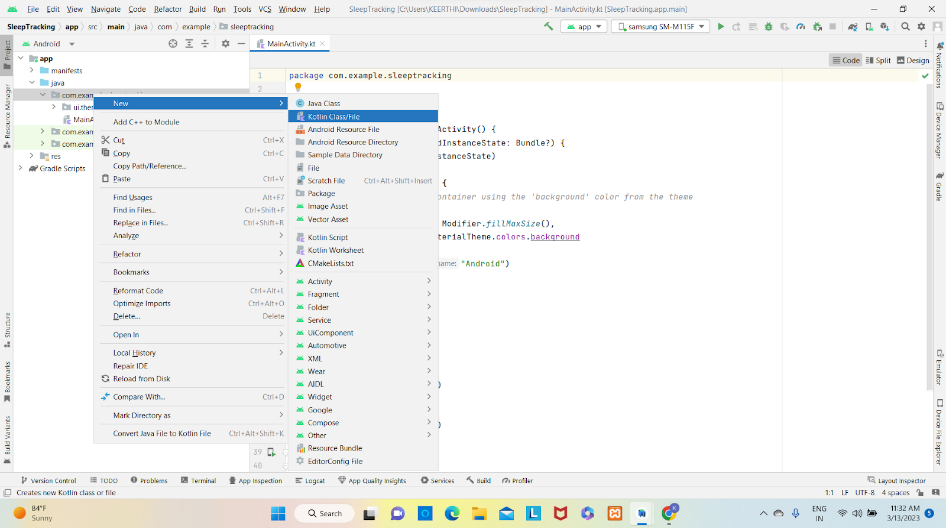


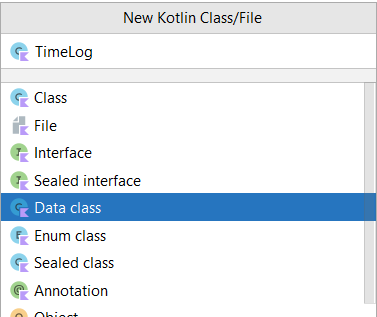
UserDatabaseHelper class code :

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/UserDatabaseHelper.kt>

### Create TimeLog Data Class

**Create TimeLog data class**



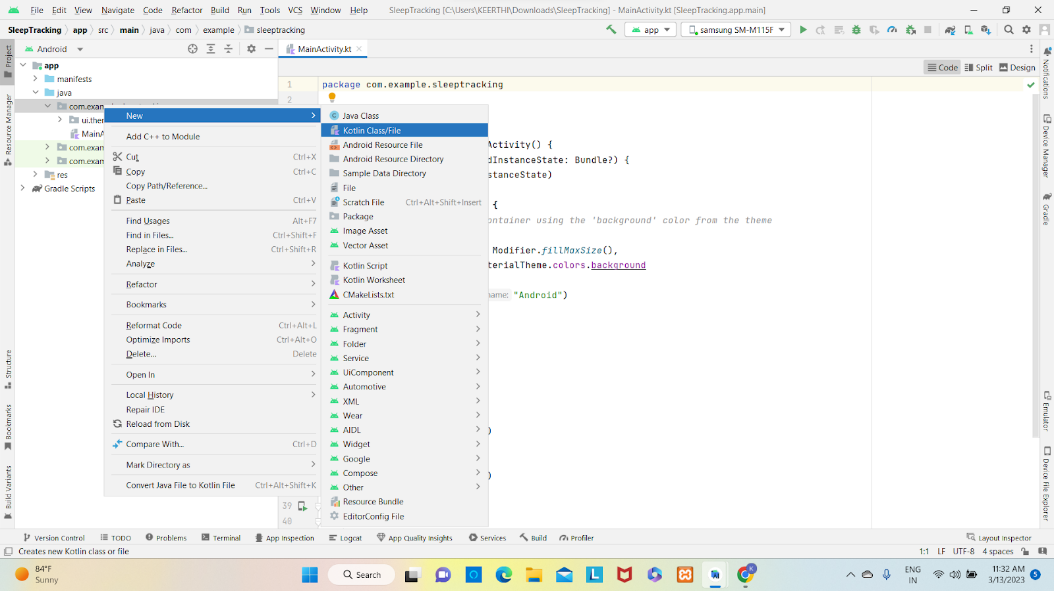


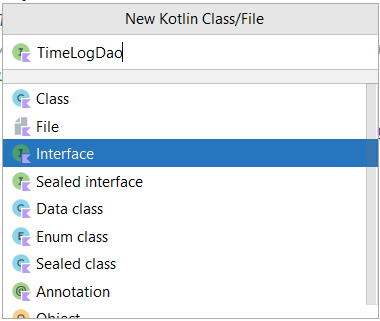
TimeLog data class code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/TimeLog.kt>

### Create An TimeLogDao Interface

**Create an TimeLogDao interface**



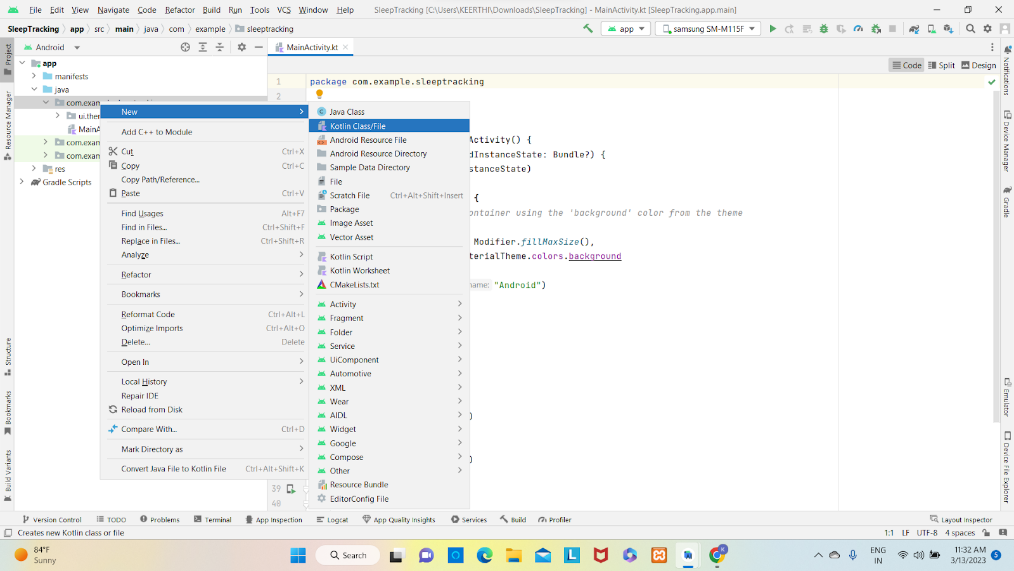


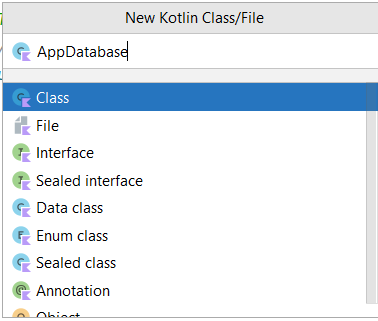
TimeLogDao interface code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/TimeLogDao.kt>

### Create An AppDatabase Class

**Create an AppDatabase class**



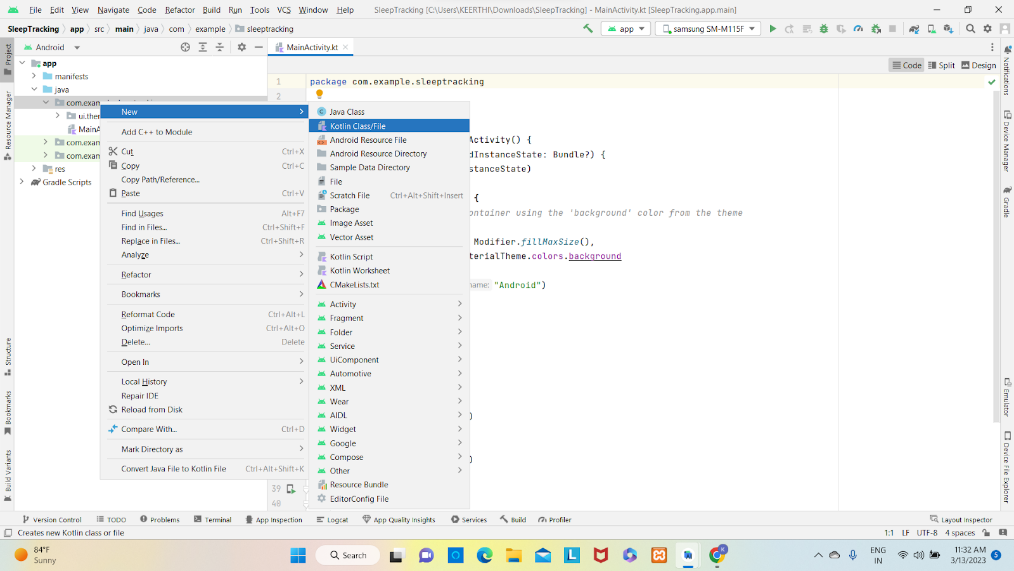


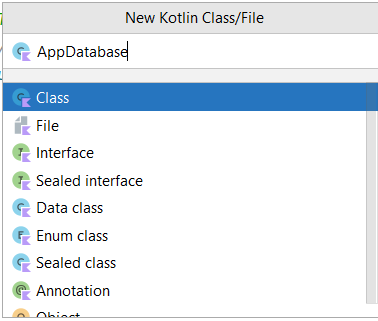
 AppDatabase class code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/AppDatabase.kt>

### Create An AppDatabase Class

**Create an AppDatabase class**



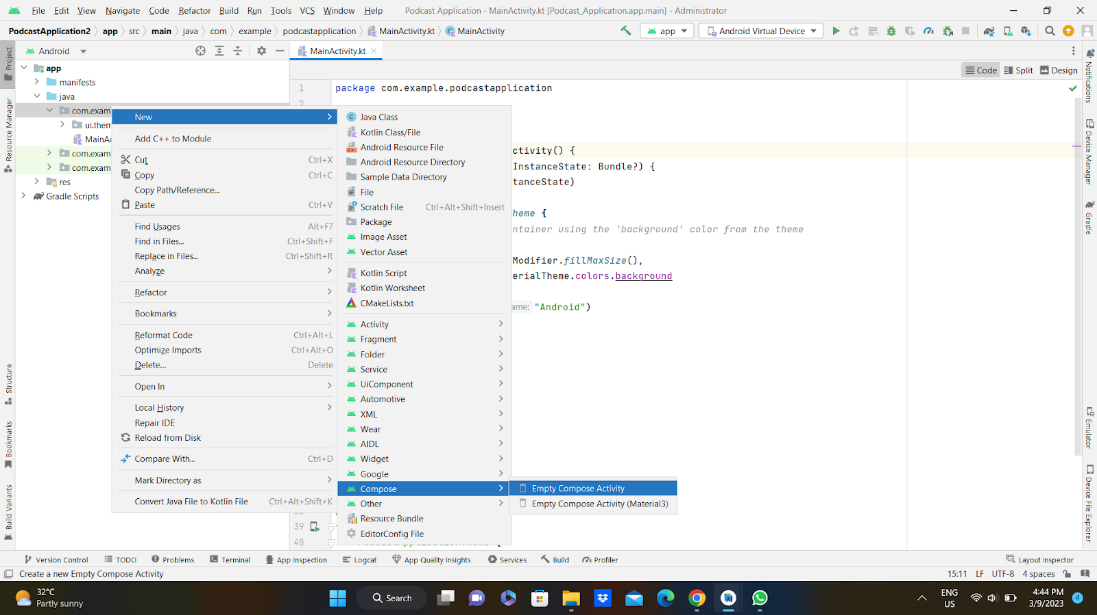


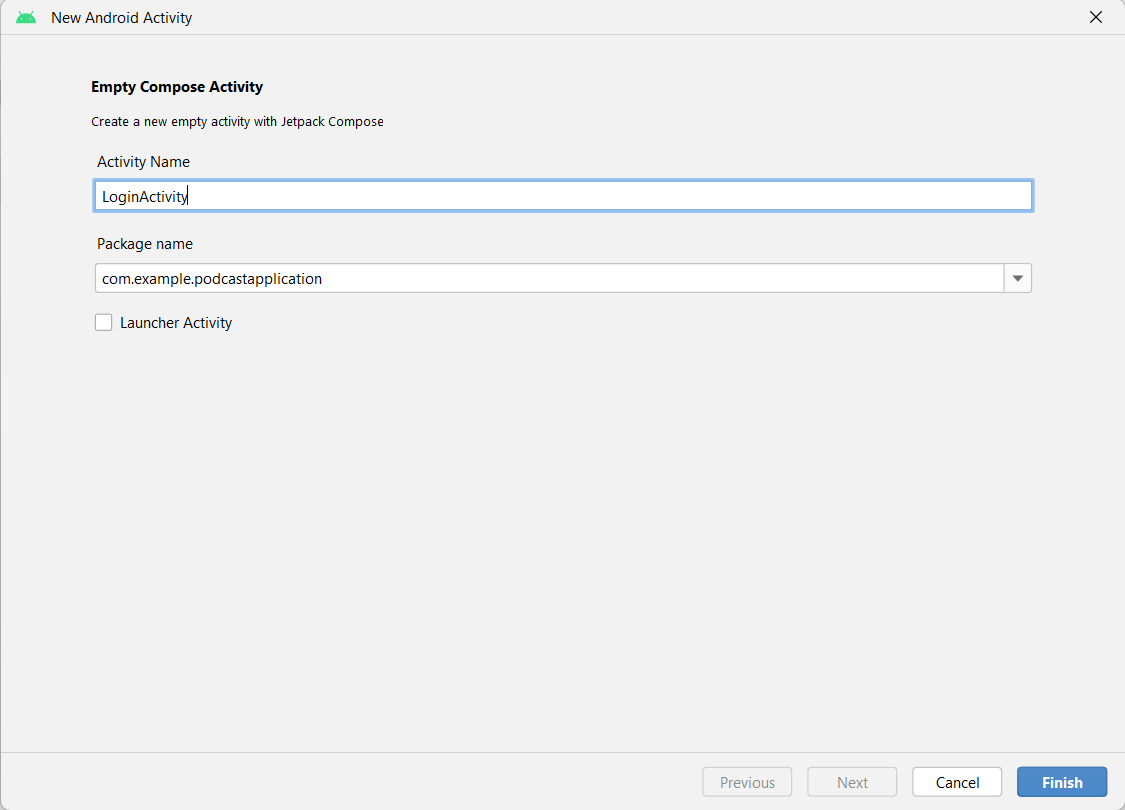
 AppDatabase class code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/AppDatabase.kt>

### Creating LoginActivity.Kt With Database

**Creating LoginActivity.kt with database**





Database connection in LoginActivity.kt

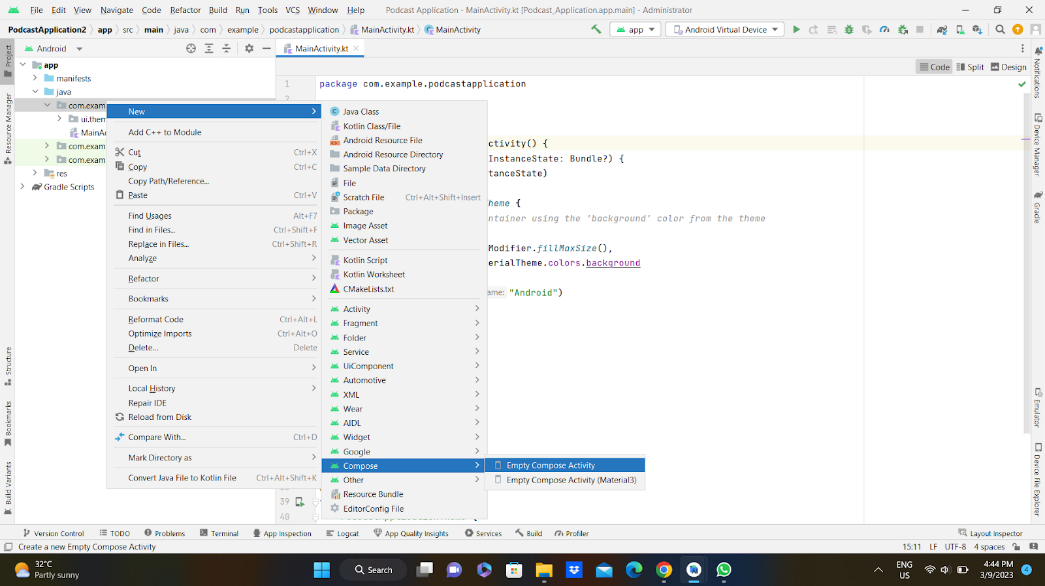


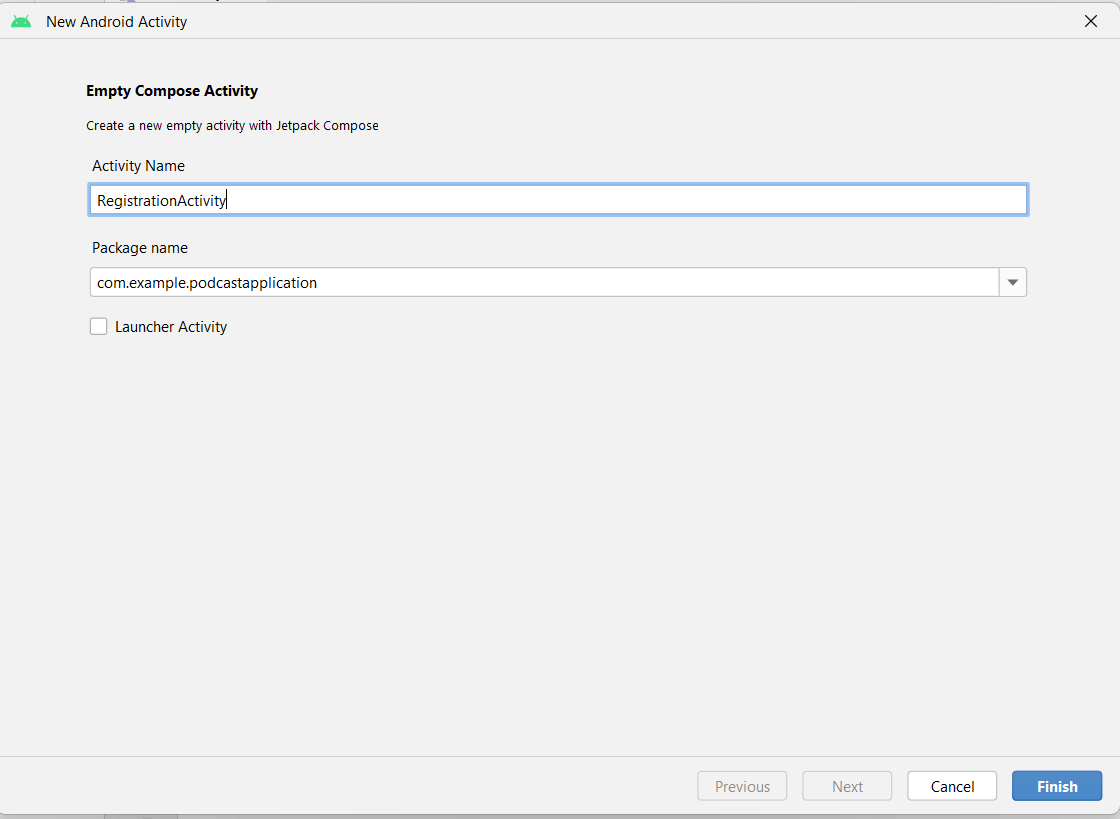
Complete code in below link:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/LoginActivity.kt>

### Creating RegistrationActivity.Kt With Database

**Creating RegistrationActivity.kt with database**





Database connection in RegistrationActivity.kt

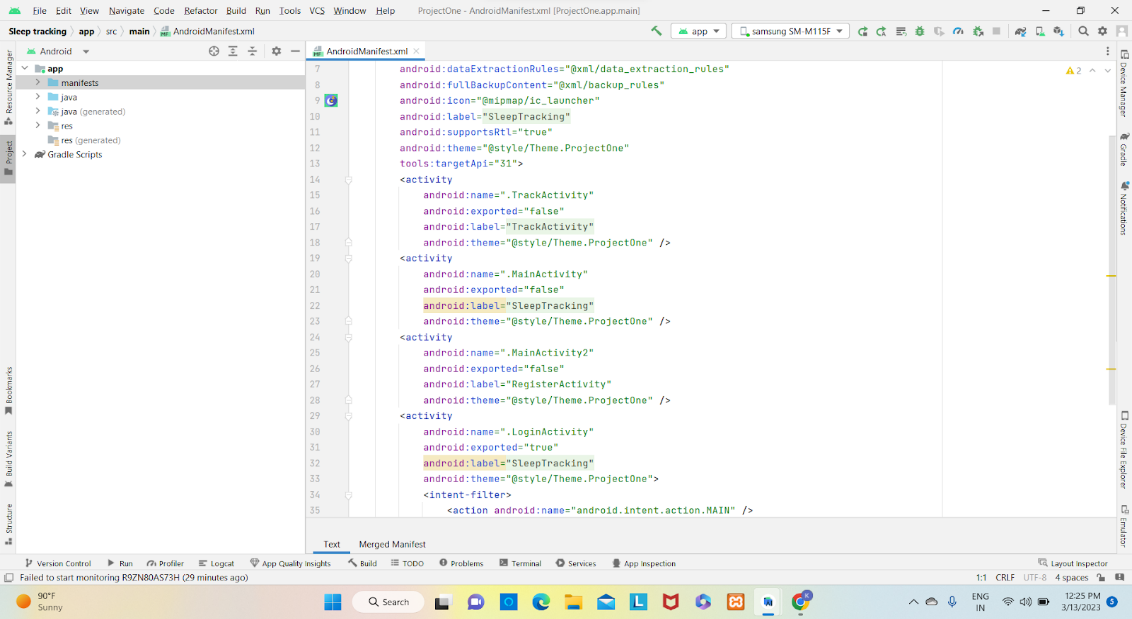


Complete code in below link:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/RegisterActivity.kt>

### Modifying AndroidManifest.Xml

**Modifying AndroidManifest.xml**



When we run the app we will get the MainActivity.kt file as our first screen , but we want LoginActivity.kt , So we need to change in AndroidManifest.xml.

Changed AndroidManifest.xml.

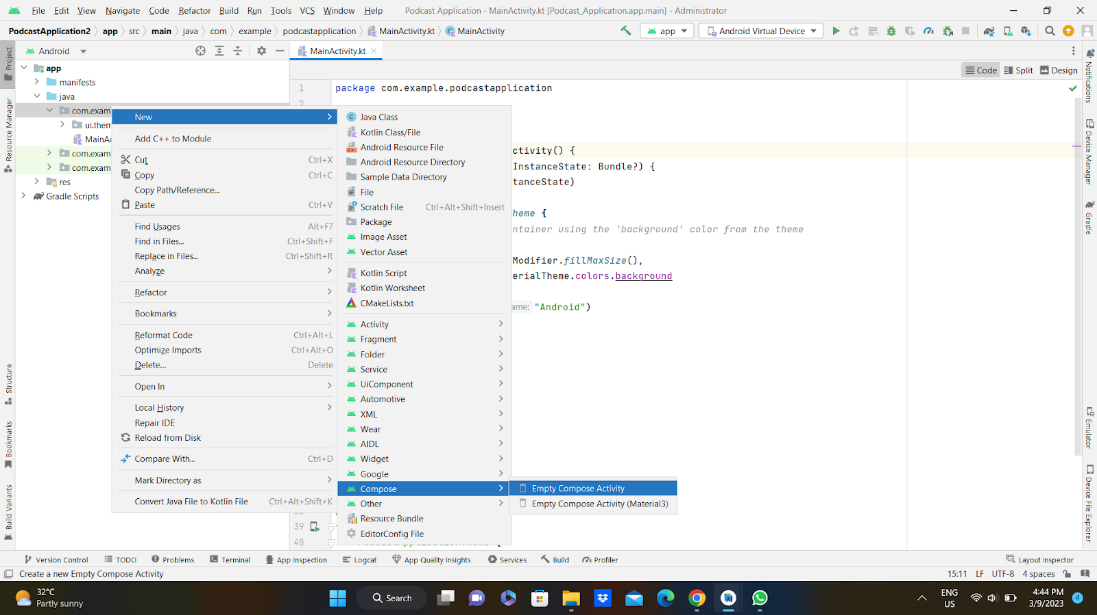


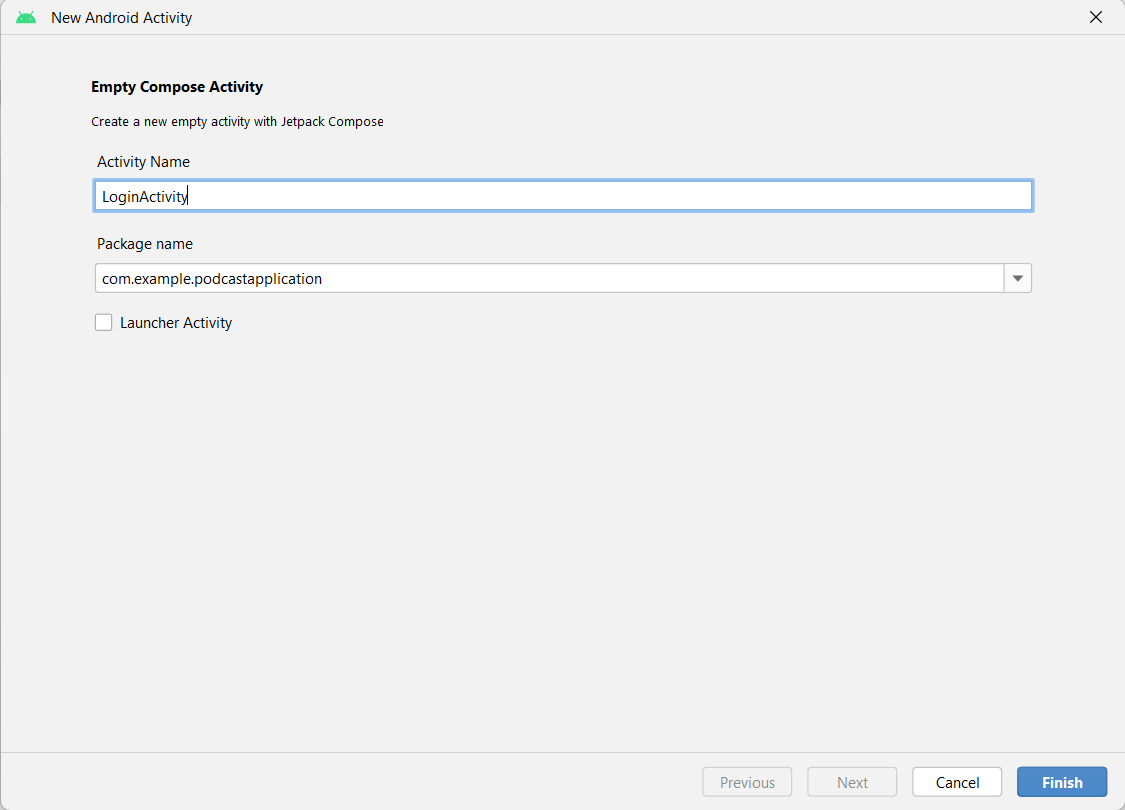
Complete AndroidManifest.xml code:

<https://github.com/smartinternz02/Sleep-tracking/blob/main/app/src/main/AndroidManifest.xml>

### Creating LoginActivity.Kt With Database

**Creating LoginActivity.kt with database**





Database connection in LoginActivity.kt

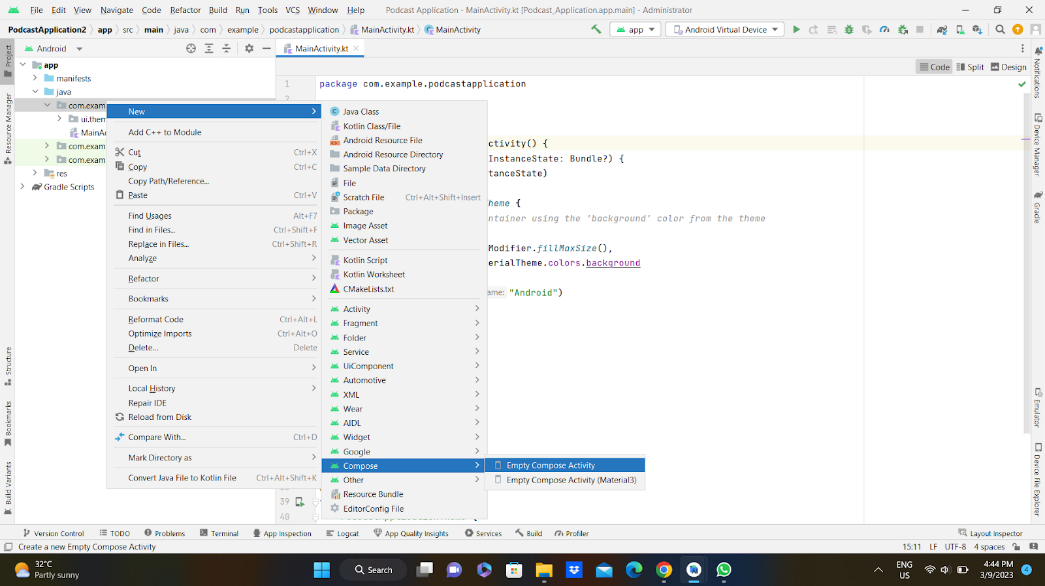


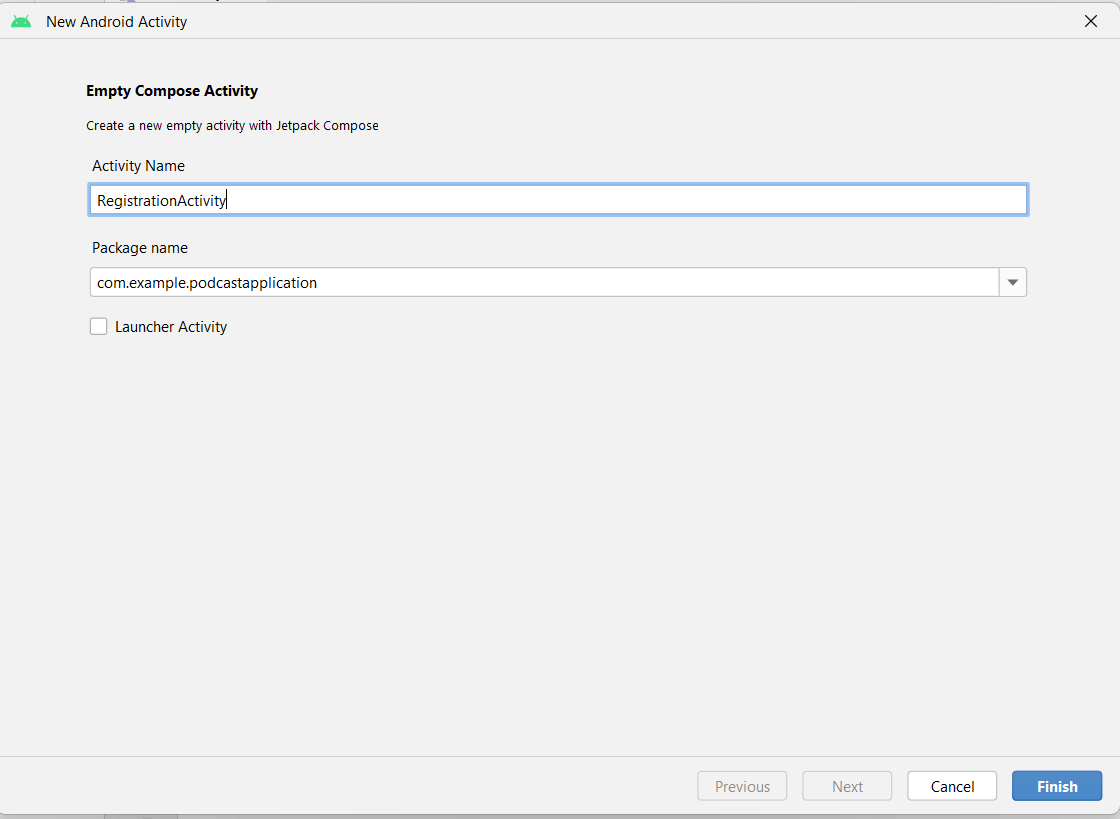
Complete code in below link:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/LoginActivity.kt>

### Creating RegistrationActivity.Kt With Database

**Creating RegistrationActivity.kt with database**



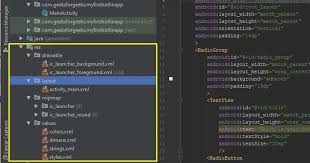


Database connection in RegistrationActivity.kt



Complete code in below link:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/RegisterActivity.kt>



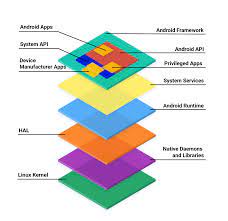
### Creating MainActivity.Kt File

**Creating MainActivity.kt file**

In MainActivity.kt file the main application is developed

* Before creating UI we need to add some images in drawables which are in res



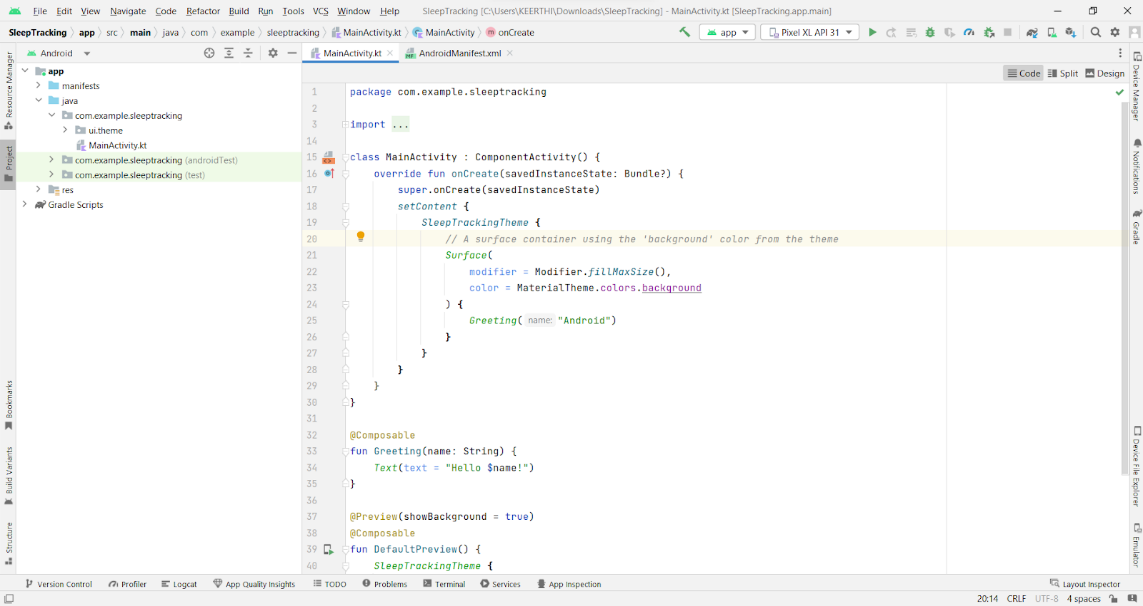


Download the required drawable from the code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/res/drawable-nodpi>

Required drawables

MainActivity.kt



Complete code in below link:

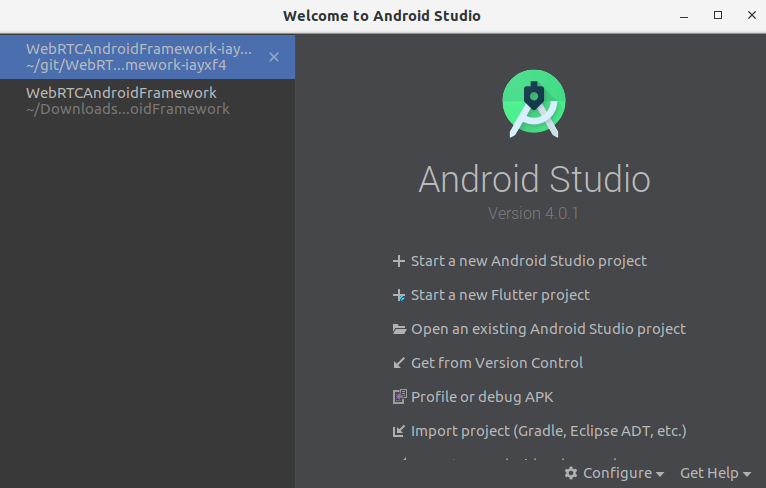
<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/MainActivity.kt>

### Creating MainActivity.Kt File

**Creating MainActivity.kt file**

In MainActivity.kt file the main application is developed

* Before creating UI we need to add some images in drawables which are in res

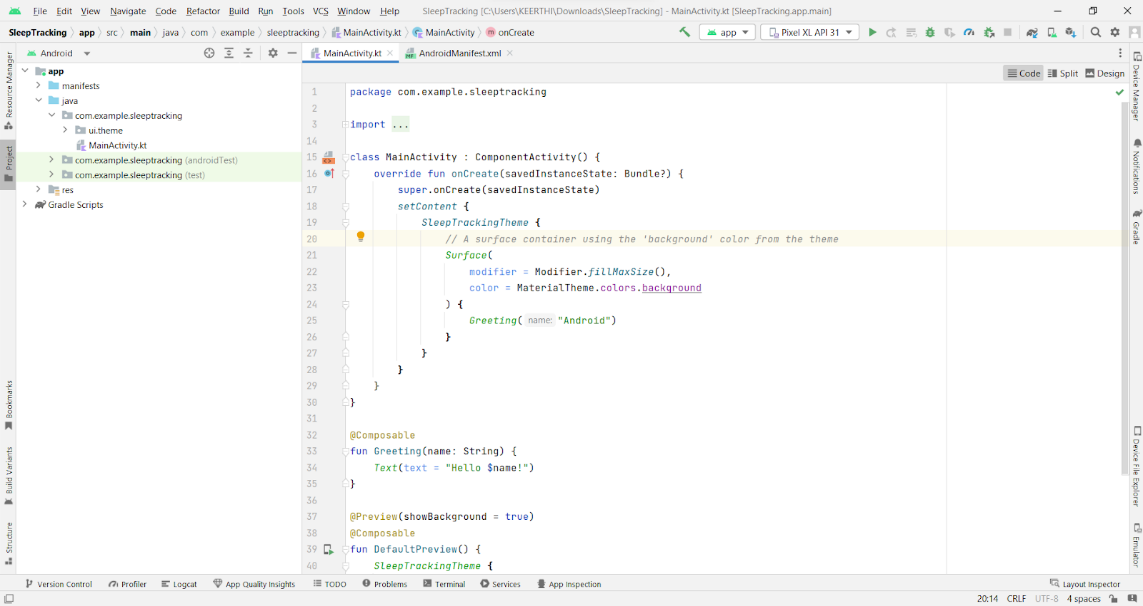


Download the required drawable from the code:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/res/drawable-nodpi>

Required drawables

MainActivity.kt

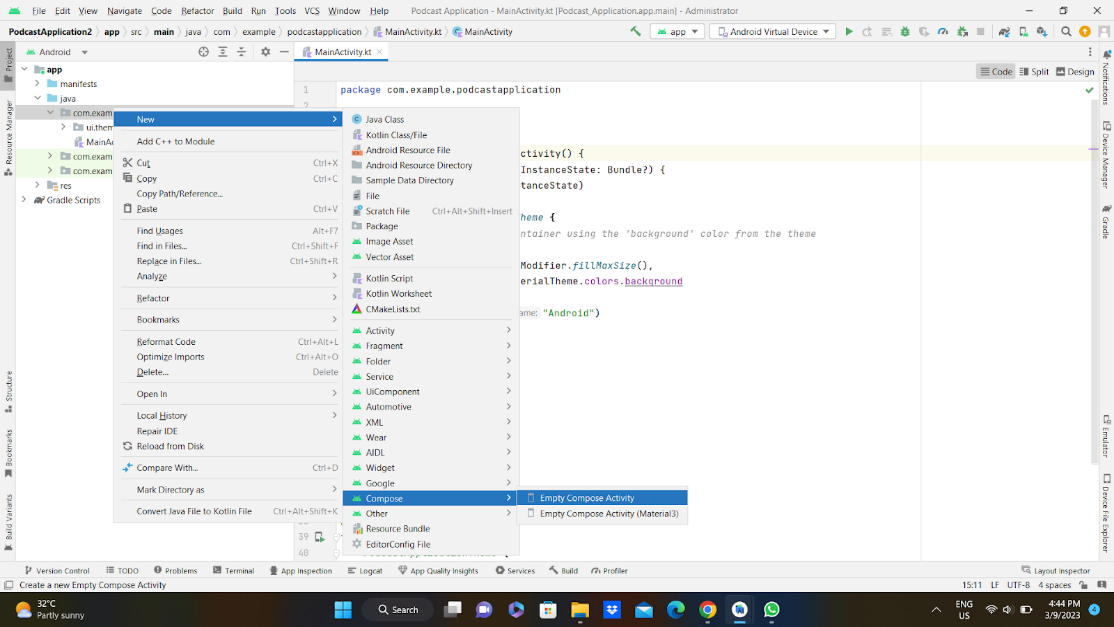


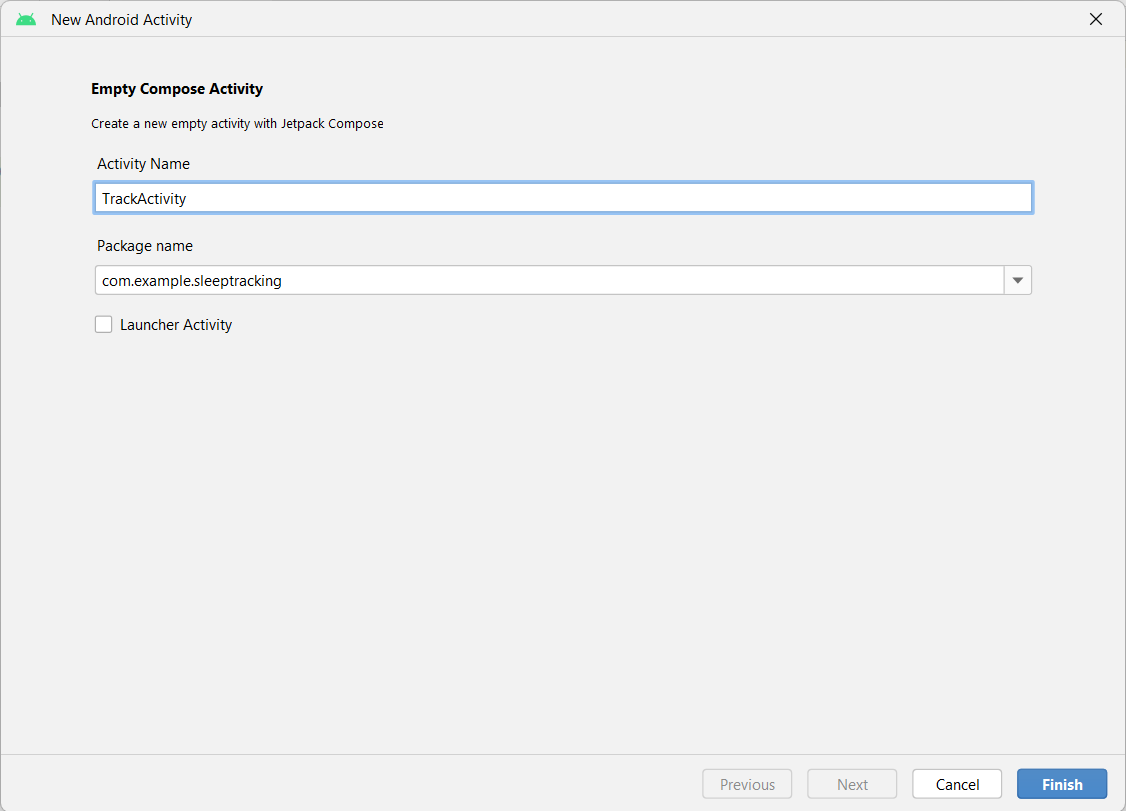
Complete code in below link:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/MainActivity.kt>

### Creating TrackActivity.Kt File

**Creating TrackActivity.kt file**





Database connection and fetching  in TrackActivity.kt

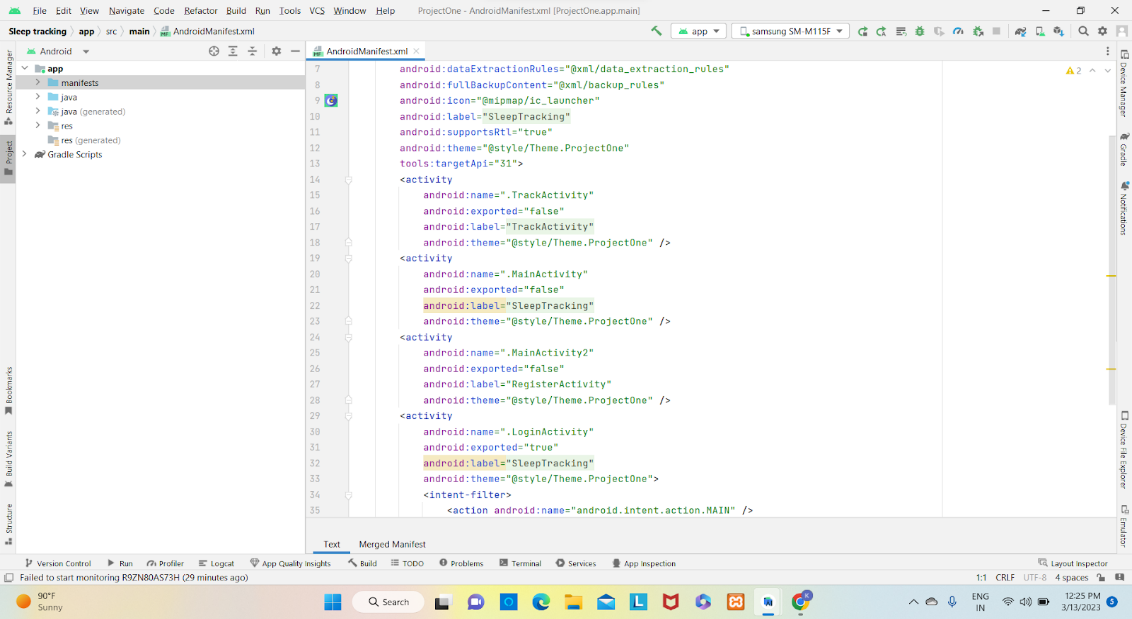


Complete code in below link:

<https://github.com/smartinternz02/Sleep-tracking/tree/main/app/src/main/java/com/example/projectone/TrackActivity.kt>

### Modifying AndroidManifest.Xml

**Modifying AndroidManifest.xml**



When we run the app we will get the MainActivity.kt file as our first screen , but we want LoginActivity.kt , So we need to change in AndroidManifest.xml.

Changed AndroidManifest.xml.



Complete AndroidManifest.xml code:

<https://github.com/smartinternz02/Sleep-tracking/blob/main/app/src/main/AndroidManifest.xml>

**Conclusion**

The network vulnerability assessment on the "altoroMutual" system conducted using

NessusEssentials reveals several vulnerabilities that need attention. Here is a summary of

the keyfindings:

**1. TLSVersion1.0ProtocolDetection(VulnerabilityID:104743):**

The remote service supports TLS version 1.0, which is considered outdated and has

knowncryptographic design flaws. Modern implementations of TLS 1.2 and 1.3 are recommended

tomitigatethesevulnerabilities.TLS1.0shouldbedisabledtoenhancesecurityandcomplywithindustryst

andards.

**2. AdditionalDNSHostnames(VulnerabilityID:46180):**

The Nessus scan detected additional DNS hostnames pointing to the remote host. It is

importanttoverifythesehostnamestoensuretheyarelegitimateanddonotposesecurityrisks.

**3. CommonPlatformEnumeration(CPE)(VulnerabilityID:45590):**

The Nessus scan enumerated CPE names that match the remote system. Understanding the

CPEinformation can help in identifying potential vulnerabilities associated with hardware

Andsoftwareproductsonthehost

**4. DeviceType(VulnerabilityID:54615):**

TheNessusscaninferredtheremotedevicetypeasa"firewall"basedontheremoteoperatingsystem

information. This helps to identify the nature of the system but does not indicate avulnerability.

**5. NessusSYNScanner(VulnerabilityID:11219):**

TheNessusscandetectedopenTCPportsontheremotehostusingSYNscanning.Whilethisinformatio

n can be useful for legitimate purposes, it should be monitored to prevent

anypotentialmisuse.

**6. NessusScanInformation(VulnerabilityID:19506):**

Details about the Nessus scan, including the version of the plugin set, the scanner edition,

andthe scan duration, were provided. This information helps in understanding the scan results

anditsconfiguration.

**7. OSIdentification(VulnerabilityID:11936):**

TheNessusscanidentifiedtheremoteoperatingsystemas"CISCOPIX7.0"usingremoteprobes.Whilethis

informationishelpfulforsystemadministrators,itdoesnotindicateanysecurityrisks.

**8.SSL/TLSVulnerabilities(VulnerabilityIDs:56984,95631,70544,10863,21643,94761,**

**156899):**

VariousSSL/TLS-relatedvulnerabilitiesweredetected,includingweakhashingalgorithmusage,known

CA SSL certificate usage, support for SSL Cipher Block Chaining, and support fordiscouraged

SSL/TLS cipher suites. These vulnerabilities can potentially compromise

theconfidentialityandintegrityofencryptedcommunications.

**4. DeviceType(VulnerabilityID:54615):**

TheNessusscaninferredtheremotedevicetypeasa"firewall"basedontheremoteoperatingsystem

information. This helps to identify the nature of the system but does not indicate avulnerability.

**5. NessusSYNScanner(VulnerabilityID:11219):**

TheNessusscandetectedopenTCPportsontheremotehostusingSYNscanning.Whilethisinformatio

n can be useful for legitimate purposes, it should be monitored to prevent

anypotentialmisuse.

**6. NessusScanInformation(VulnerabilityID:19506):**

Details about the Nessus scan, including the version of the plugin set, the scanner edition,

andthe scan duration, were provided. This information helps in understanding the scan results

anditsconfiguration.

**7. OSIdentification(VulnerabilityID:11936):**

TheNessusscanidentifiedtheremoteoperatingsystemas"CISCOPIX7.0"usingremoteprobes.Whilethis

informationishelpfulforsystemadministrators,itdoesnotindicateanysecurityrisks.

**8.SSL/TLSVulnerabilities(VulnerabilityIDs:56984,95631,70544,10863,21643,94761,**

**156899):**

VariousSSL/TLS-relatedvulnerabilitiesweredetected,includingweakhashingalgorithmusage,known

CA SSL certificate usage, support for SSL Cipher Block Chaining, and support fordiscouraged

SSL/TLS cipher suites. These vulnerabilities can potentially compromise

theconfidentialityandintegrityofencryptedcommunications.