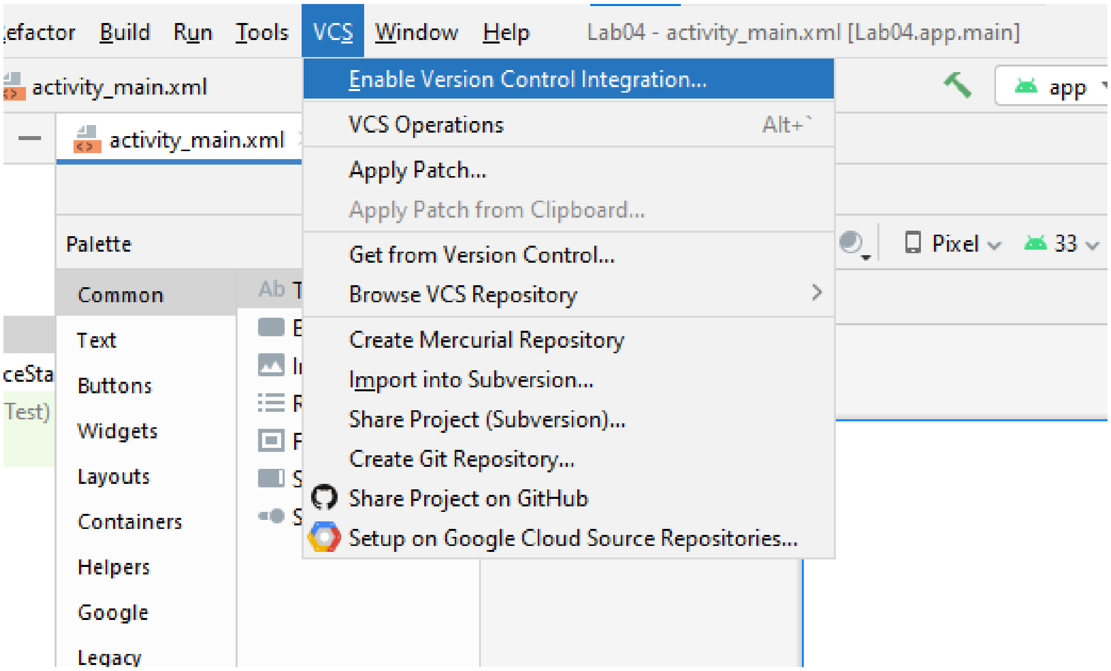
IT2010 – Mobile Application Development
BSc (Hons) in Information Technology
2nd Year
Faculty of Computing
SLIIT

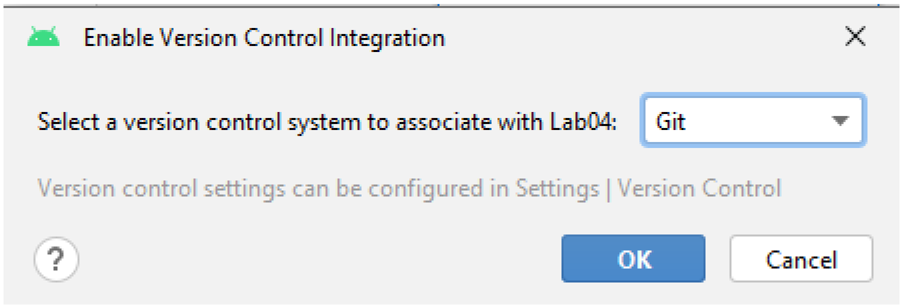
2025 – Lab 02

Version Controlling

Initializing Git and creating new branch

1. Enabling version control in Android studio

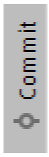

efactor
Build
Run
Tools
VCS
Window
Help
Lab04 - activity_main.xml [Lab04.app.main]
activity_main.xml
app
activity_main.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Pixel
33
Enable Version Control Integration...
VCS Operations
Alt+`
Apply Patch...
Apply Patch from Clipboard...
Get from Version Control...
Browse VCS Repository
Create Mercurial Repository
Import into Subversion...
Share Project (Subversion)...
Create Git Repository...
Share Project on GitHub
Setup on Google Cloud Source Repositories...



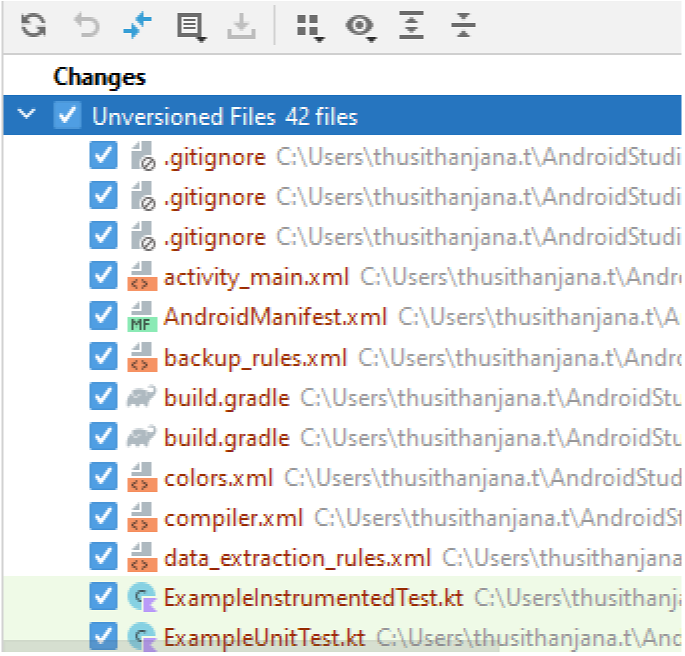
Select a version control system to associate with Lab04: Git

Version control settings can be configured in Settings | Version Control

2. Select commit from the Left side panel



3. Click Unversioned Files

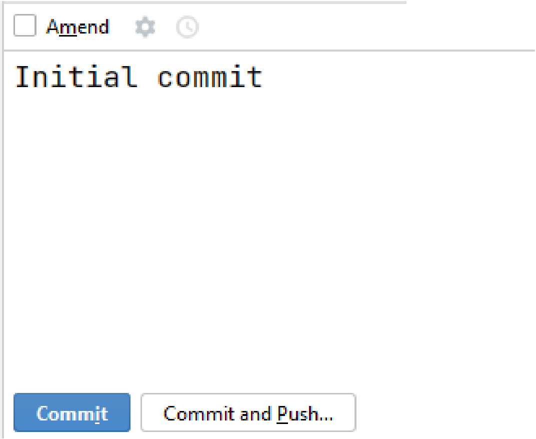


Changes

☑ Unversioned Files 42 files

- ☑ .gitignore C:\Users\thusithanjana.t\AndroidStudi
- ☑ .gitignore C:\Users\thusithanjana.t\AndroidStudi
- ☑ .gitignore C:\Users\thusithanjana.t\AndroidStudi
- ☑ activity_main.xml C:\Users\thusithanjana.t\Andr
- ☑ AndroidManifest.xml C:\Users\thusithanjana.t\A
- ☑ backup_rules.xml C:\Users\thusithanjana.t\Andr
- ☑ build.gradle C:\Users\thusithanjana.t\AndroidStu
- ☑ build.gradle C:\Users\thusithanjana.t\AndroidStu
- ☑ colors.xml C:\Users\thusithanjana.t\AndroidStud
- ☑ compiler.xml C:\Users\thusithanjana.t\AndroidSt
- ☑ data_extraction_rules.xml C:\Users\thusithanjana
- ☑ ExampleInstrumentedTest.kt C:\Users\thusithanj
- ☑ ExampleUnitTest.kt C:\Users\thusithanjana.t\And

4. Add a proper commit message and click commit

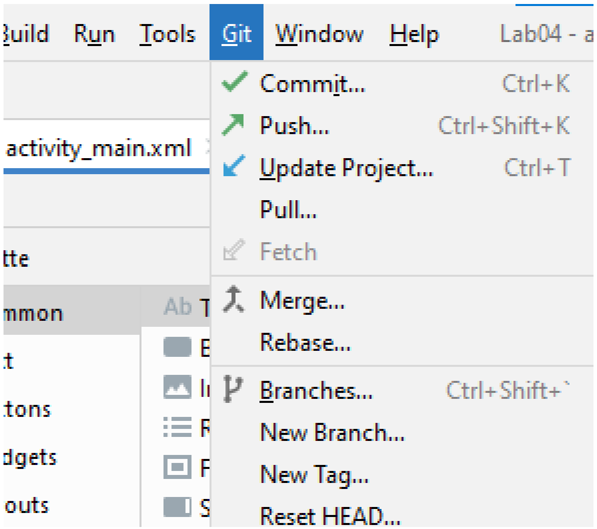


☐ Amend

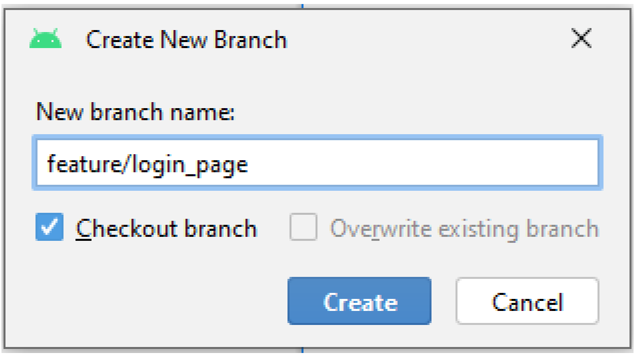
```
Initial commit
```

5. There will be warnings. Click the button commit anyway

6. Click git and select new branch



7. Create a branch named “feature/login_page”

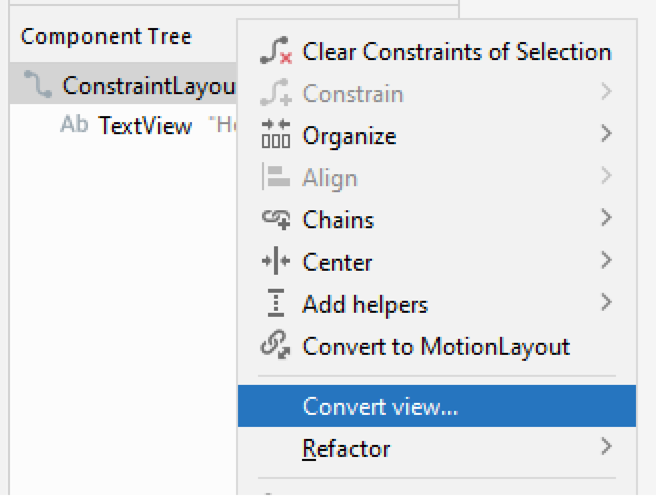


New branch name: feature/login_page

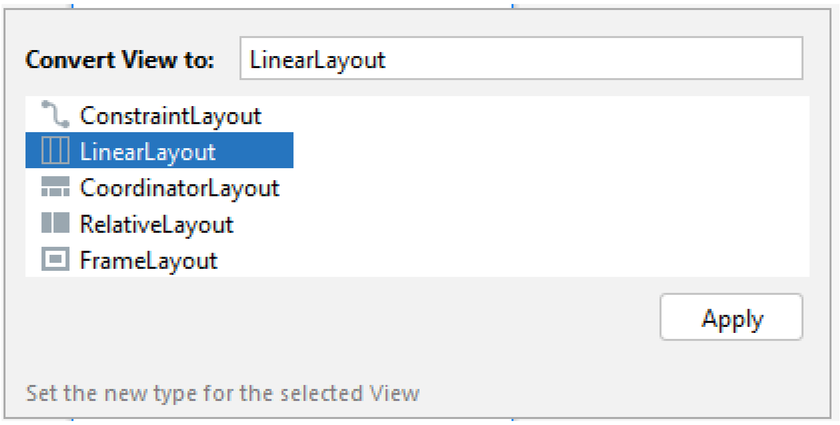
☑ Checkout branch ☐ Overwrite existing branch

Introduction to Linear Layout

1. Right Click the Constraint layout from the component tree and select Convert View

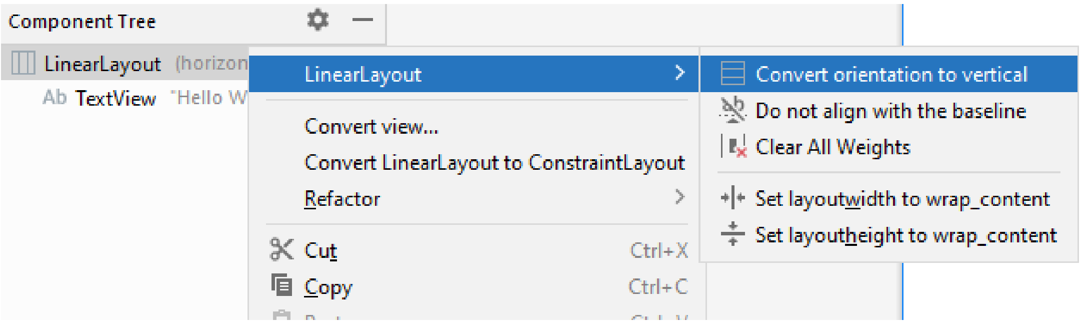


2. Then Select Linear Layout

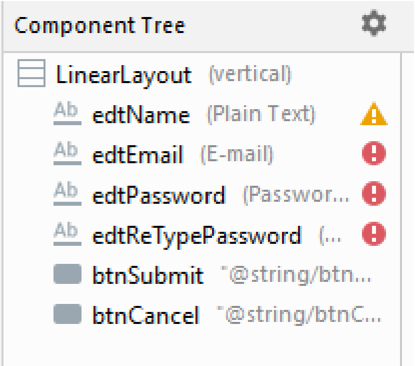

Convert View to: LinearLayout

Set the new type for the selected View

3. Convert the horizontal layout to vertical layout

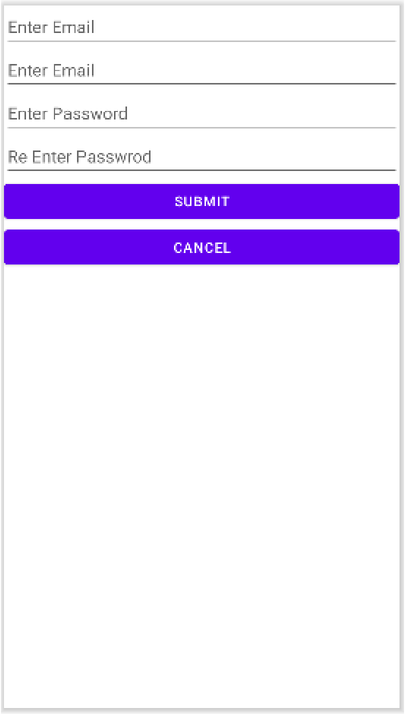

Component Tree
LinearLayout (horizon
Ab TextView "Hello W
LinearLayout
Convert view...
Convert LinearLayout to ConstraintLayout
Refactor
Cut Ctrl+X
Copy Ctrl+C
Convert orientation to vertical
Do not align with the baseline
Clear All Weights
Set layoutwidth to wrap_content
Set layoutheight to wrap_content

4. Design the following User interface
    a. Add each relevant component by dragging from the palette to the screen respectively
    b. Add all the necessary strings to the resources
    c. Properly id all the views



Component Tree

- LinearLayout (vertical)
    - edtName (Plain Text) ⚠
    - edtEmail (E-mail) ❗
    - edtPassword (Passwor... ❗
    - edtReTypePassword (... ❗
    - btnSubmit "@string/btn...
    - btnCancel "@string/btnC...



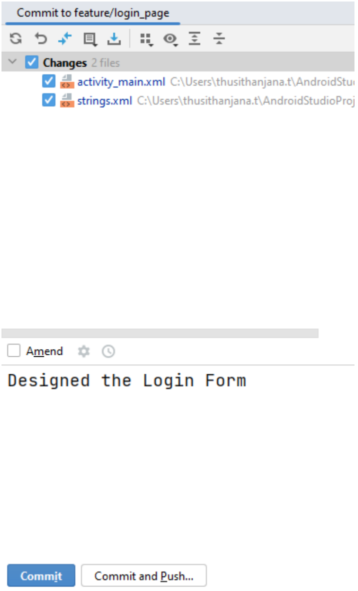
Enter Email

Enter Email

Enter Password

Re Enter Passwrod

5. Select all the input fields and set the layout_height to 48dp
6. Select all the component and give 16dp margin (Top, Bottom, Left, Right)
7. Commit the changes to the repository



Commit to feature/login_page

☑ Changes 2 files
    ☑ activity_main.xml C:\Users\thusithanjana.t\AndroidStu
    ☑ strings.xml C:\Users\thusithanjana.t\AndroidStudioProj

□ Amend

```
Designed the Login Form
```

Introduction to GitHub

GitHub is a web-based platform that is primarily used for version control and collaborative software development. It provides a platform for hosting and sharing code repositories, as well as a range of tools and features for managing and collaborating on code projects.

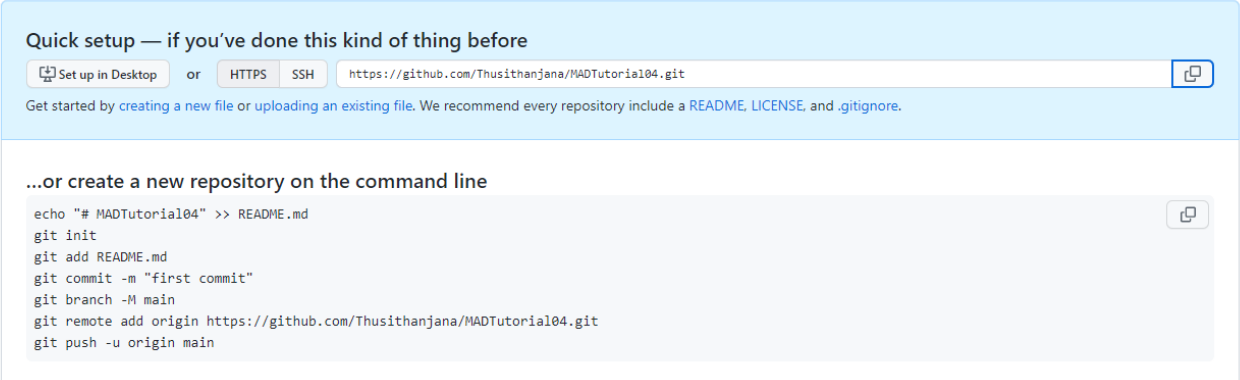
GitHub allows developers to create, fork, and contribute to open-source projects. It provides tools for code review, bug tracking, and documentation, and allows developers to work together on projects from anywhere in the world.

GitHub is also widely used for personal and professional projects, ranging from small personal projects to large-scale enterprise applications. It has become an important tool for developers to collaborate and share knowledge and is used by millions of developers and organizations around the world.

1. Create a GitHub account

[GitHub: Let’s build from here · GitHub]

2. Create a new repository. Name it ‘MADTutorial04’
3. Make sure to keep the other fields empty
4. Copy the link for the GitHub repository



Quick setup — if you’ve done this kind of thing before

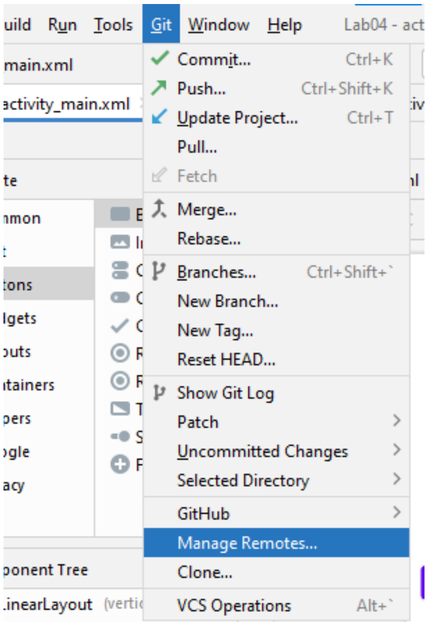
https://github.com/Thusithanjana/MADTutorial04.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

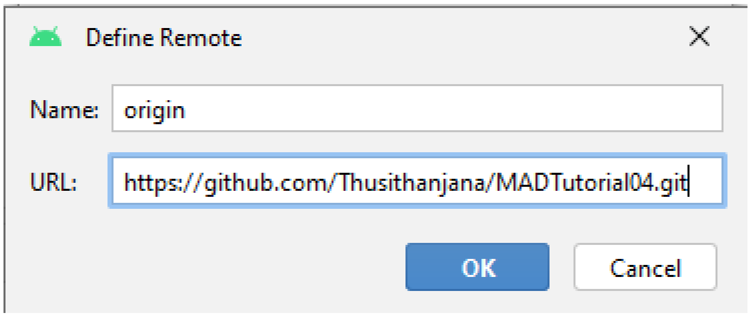
…or create a new repository on the command line

```
echo "# MADTutorial04" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Thusithanjana/MADTutorial04.git
git push -u origin main
```

5. Come back to Android Studio. Select Git and click mange remotes



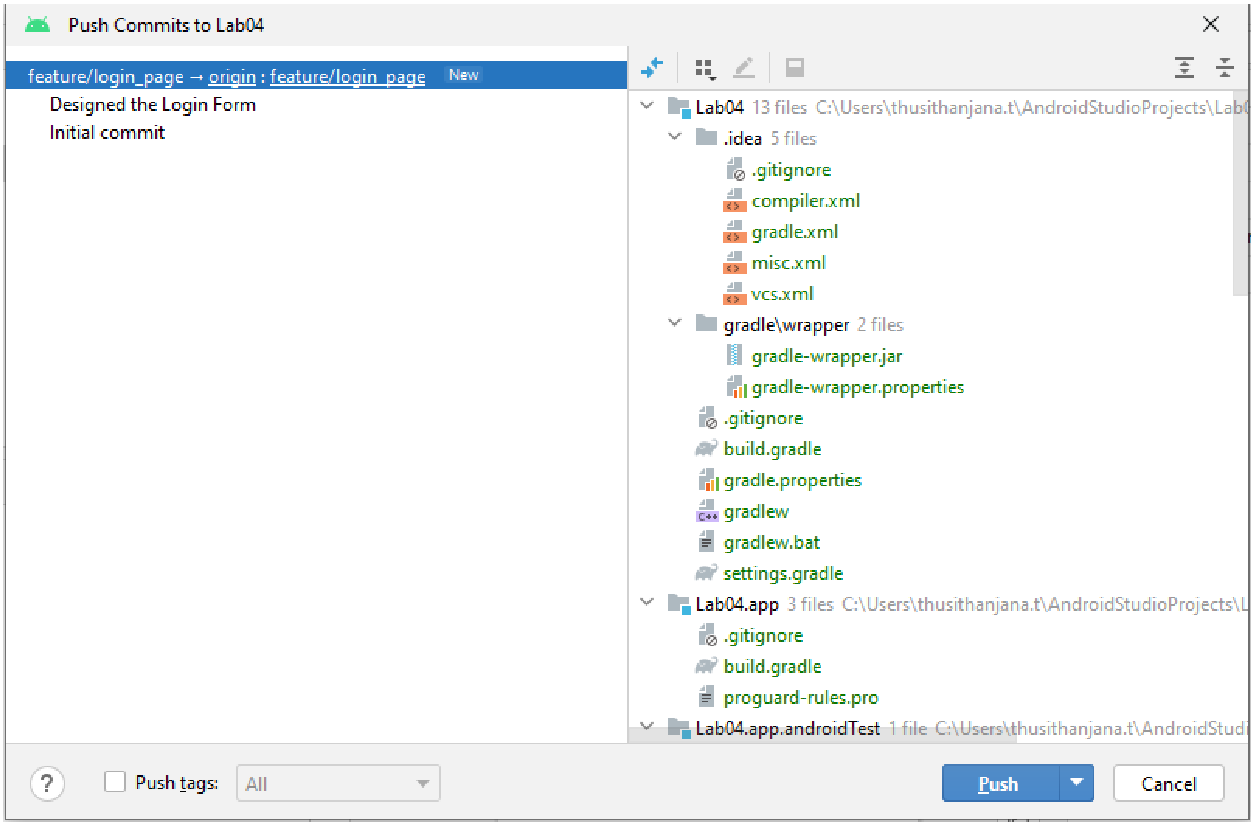
6. Click the + button and paste the link to the displayed dialog box. And click ok



Name: origin

URL: https://github.com/Thusithanjana/MADTutorial04.git

7. Then click git and select Push.
8. After that from the popped window click the push button.
9. For that you must login to GitHub from you PC. Provide the credentials for it.



Push Commits to Lab04

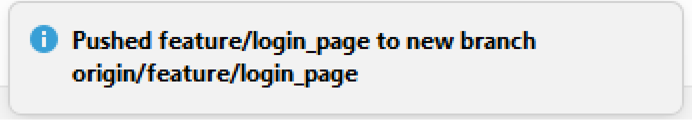
feature/login_page → origin : feature/login_page New

Designed the Login Form

Initial commit

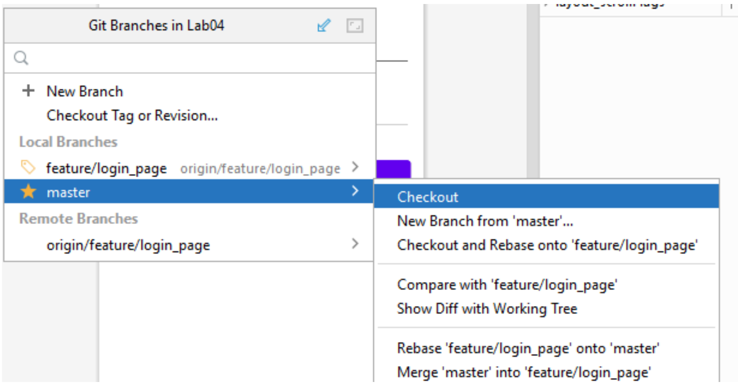
```
Lab04 13 files C:\Users\thusithanjana.t\AndroidStudioProjects\Lab0
  .idea 5 files
    .gitignore
    compiler.xml
    gradle.xml
    misc.xml
    vcs.xml
  gradle\wrapper 2 files
    gradle-wrapper.jar
    gradle-wrapper.properties
  .gitignore
  build.gradle
  gradle.properties
  gradlew
  gradlew.bat
  settings.gradle
Lab04.app 3 files C:\Users\thusithanjana.t\AndroidStudioProjects\L
  .gitignore
  build.gradle
  proguard-rules.pro
Lab04.app.androidTest 1 file C:\Users\thusithanjana.t\AndroidStudi
```

10. If the push is success following message will be displayed



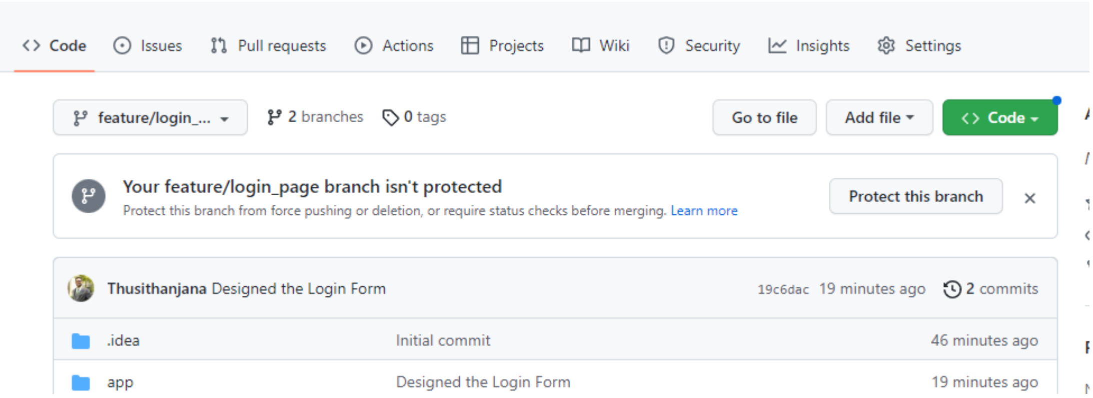
Pushed feature/login_page to new branch origin/feature/login_page

11. After that go to the GitHub repository and refresh the page
12. Go back to Android studio. Click Git and select Branches
13. Then select the master branch and click checkout



Git Branches in Lab04
+ New Branch
Checkout Tag or Revision...
Local Branches
feature/login_page origin/feature/login_page
master
Remote Branches
origin/feature/login_page
Checkout
New Branch from 'master'...
Checkout and Rebase onto 'feature/login_page'
Compare with 'feature/login_page'
Show Diff with Working Tree
Rebase 'feature/login_page' onto 'master'
Merge 'master' into 'feature/login_page'

14. Then you can see the initial project. Push that to the GitHub as well
15. Refresh the GitHub page and you can see there are two branches.
16. From GitHub, select Pull Requests



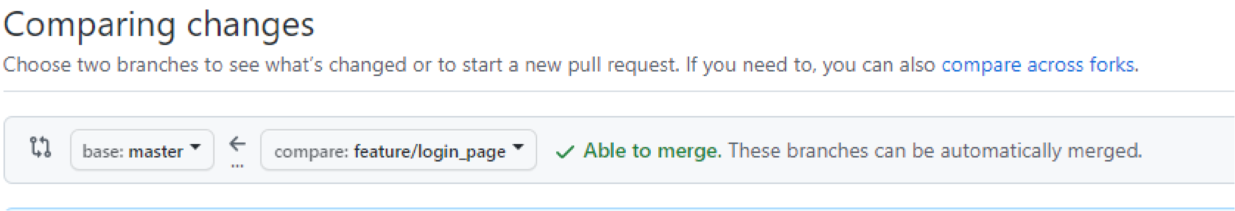
Your feature/login_page branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. Learn more

Thusithanjana Designed the Login Form 19c6dac 19 minutes ago 2 commits

.idea	Initial commit	46 minutes ago
app	Designed the Login Form	19 minutes ago

17. Click new pull request.
18. Select the branches as follows

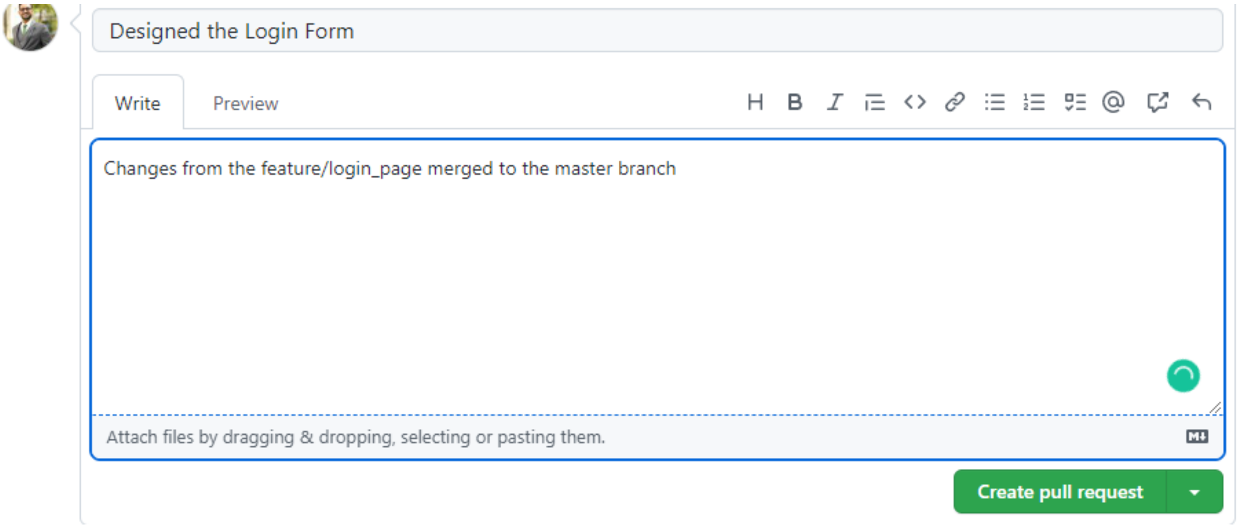


Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: master ← compare: feature/login_page ✓ Able to merge. These branches can be automatically merged.

19. And Click Create Pull Request
20. Then add a proper comment and click create pull request

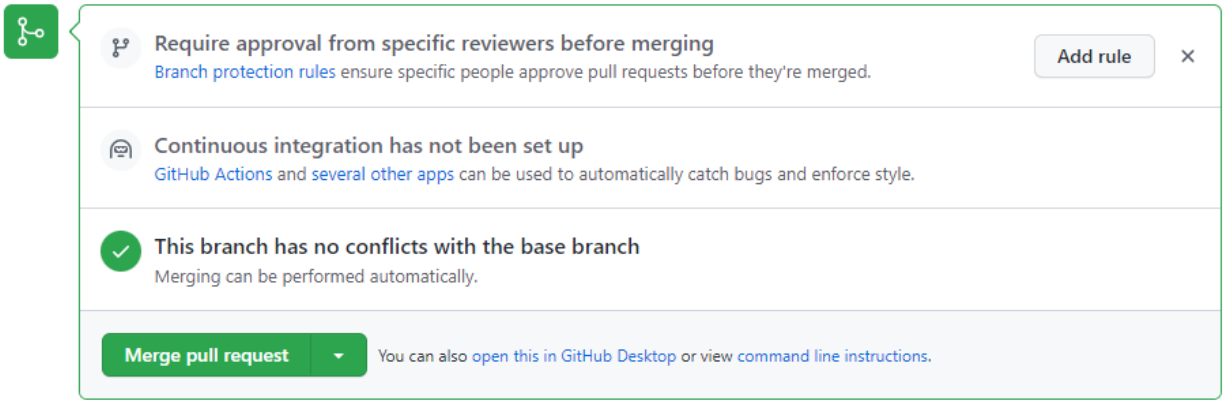


Designed the Login Form

Changes from the feature/login_page merged to the master branch

Attach files by dragging & dropping, selecting or pasting them.

21. Then click Merge Pull Request. And then confirm



Require approval from specific reviewers before merging

Branch protection rules ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up

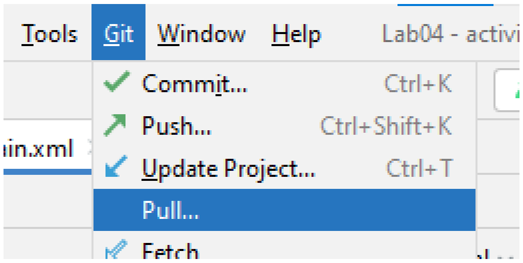
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

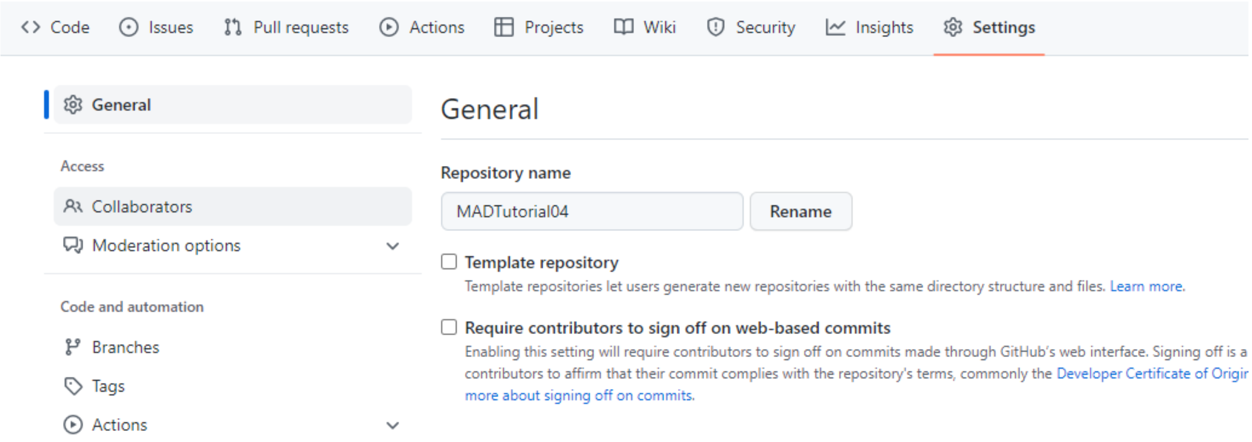
22. Check the code in the master branch form the GitHub
23. Go back to the Android Studio. Click Git and select pull.



24. Then select the branch as master and click pull.
25. Observe the changes.

GitHub Collaboration

1. Go to the GitHub repository
2. Select settings
3. Select the Collaborators tab from the left side panel



General

Repository name

MADTutorial04

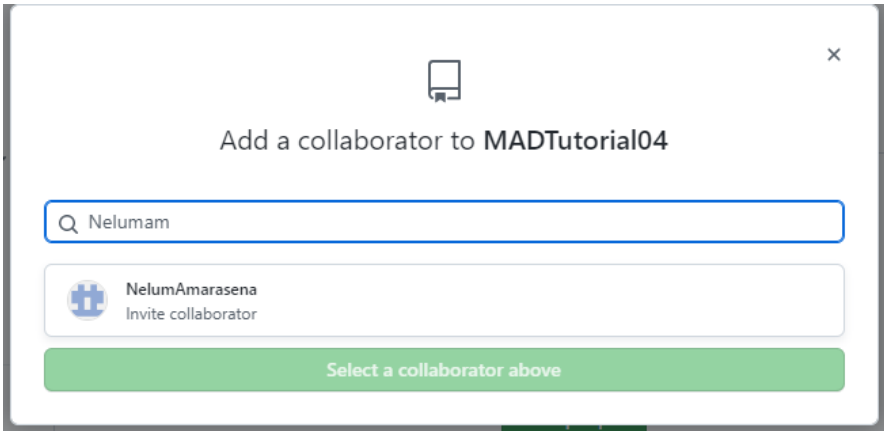
□ Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more.

□ Require contributors to sign off on web-based commits

Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origir more about signing off on commits.

4. Click the 'Add People' Button to invite collaborators.
5. Add the GitHub user name of someone next you in the lab



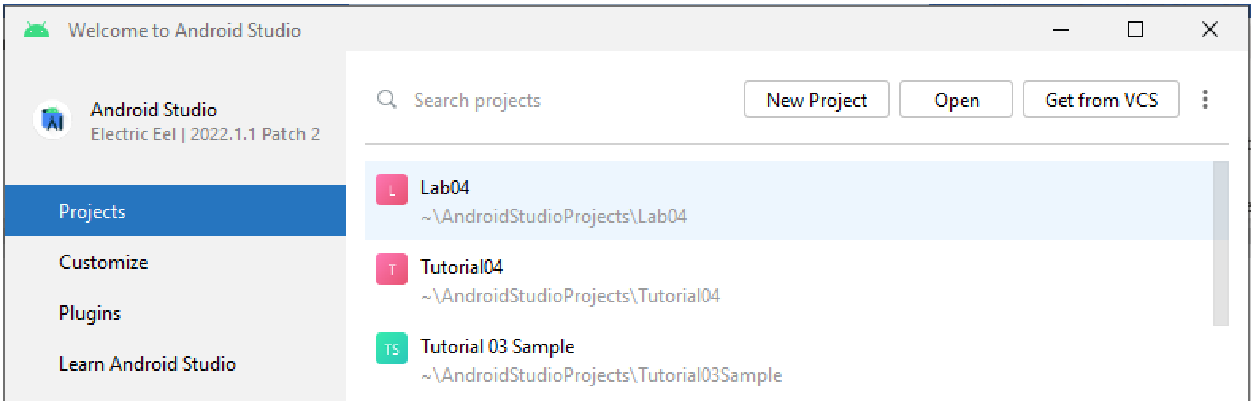
Add a collaborator to **MADTutorial04**

Nelumam

NelumAmarasena
Invite collaborator

6. Ask the other member to accept the request.
7. You can accept the other members request as well.
8. Then close the current Android Project
9. You should be able to see the Welcome screen

10. Select get from VCS


Welcome to Android Studio
Android Studio
Electric Eel | 2022.1.1 Patch 2
Projects
Customize
Plugins
Learn Android Studio
Search projects
New Project
Open
Get from VCS
L
Lab04
~\AndroidStudioProjects\Lab04
T
Tutorial04
~\AndroidStudioProjects\Tutorial04
TS
Tutorial 03 Sample
~\AndroidStudioProjects\Tutorial03Sample

11. Add the URL of the neighbor’s repository
12. Click clone
13. Once the project loads, create a new branch name ‘feature/home_screen’
14. Design the home screen by adding some text views
15. Commit the changes
16. Push the feature/home_screen to your neighbor’s repository
17. At this stage your neighbor should be able to do the same thing and you should be able to see the new branch from your git repository
18. Repeat the steps for merging branches to the master branch.