

Lab 14: Introduction to GitHub Cloning Repository

Lab 13: Introduction to GitHub and Cloning Repository

Objective

1. Understand the fundamentals of Git as a version control system.
2. Learn how to connect local projects to GitHub.
3. Practice committing changes, pushing code, pulling remote changes, and resolving conflicts.
4. Understand repository structure, staging, committing, branching, and remote synchronization.
5. Explore collaborative workflows used in software development teams.

Reference Material

- Lecture Notes: Introduction to Version Control
- Textbook: *Practical Git and GitHub*
- Classic Reference: *Pro Git* (Scott Chacon)

Latest References (2024–2025):

- GitHub Docs: *Getting Started with GitHub*
- Atlassian Git Tutorial
- Git Documentation – Command Reference

Activity Timeline

Activity Name	Time
Introduction & Overview (Git, GitHub, Workflow)	30 minutes
Walkthrough Task 1: Cloning + Project Setup	40 minutes
Walkthrough Task 2: Commit, Push & Pull Changes	50 minutes
Practice Tasks (Git Workflow Scenarios)	40 minutes
Conclusion & Discussion (Collaboration & CI/CD)	20 minutes
Total	180 minutes

Introduction:

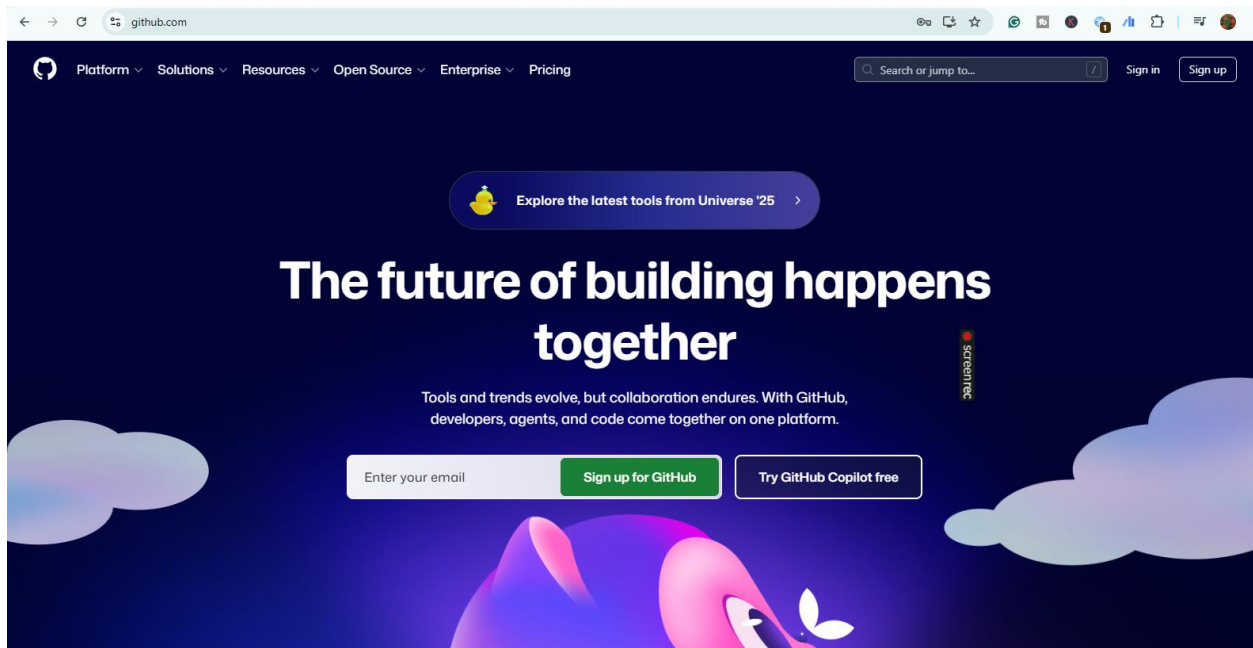
Version control is an essential component of modern software development. Git allows developers to track changes, maintain revisions, and collaborate efficiently across distributed environments.

GitHub is the most widely used hosting platform for Git repositories. It provides online repository management, collaboration features, pull requests, issue tracking, reviews, and integration pipelines.

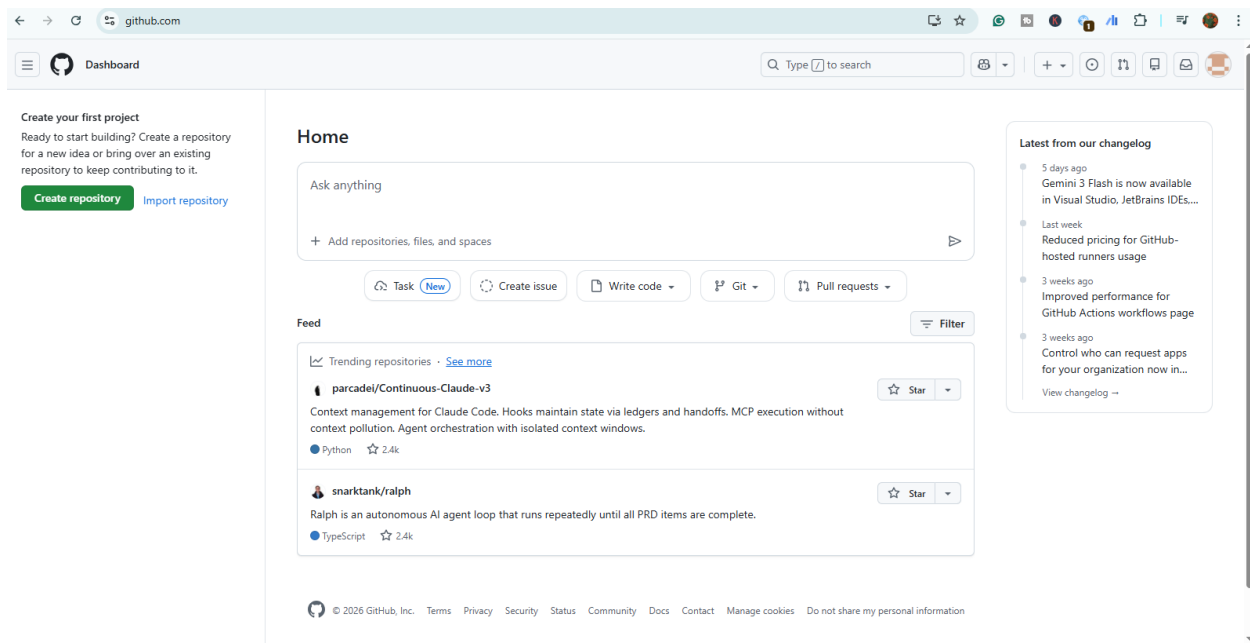
In this lab, students will learn how to work with Git and GitHub, manage code versions, commit changes, and synchronize local projects with remote repositories.

Walkthrough Task 1:

Create account on Github



The dashboard of github will look like below



Create a repository

1 General

Owner *

 hafuumahmood-hub

Repository name *

/ SCD Lab 14

✓ Your new repository will be created as SCD-Lab-14.

The repository name can only contain ASCII letters, digits, and the characters `.`, `-`, and `_`.

Great repository names are short and memorable. How about [crispy-meme](#)?

Description


This will cover the basics of how to create the repository and commit changes in it

83 / 350 characters

2 Configuration

Choose visibility *

Choose who can see and commit to this repository

 Public

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

Off ☐

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

Create repository

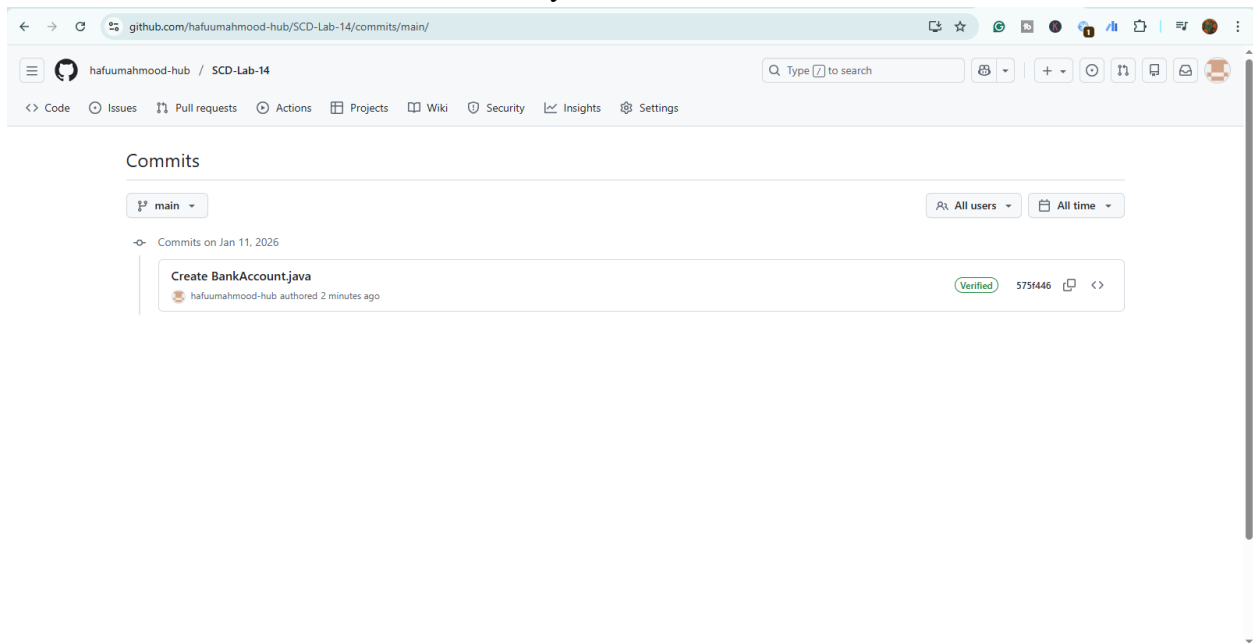
After that, invite the collaborators (teammates that can make changes to your project)

The screenshot shows the 'Collaborators and teams' settings for a GitHub repository. On the left is a sidebar with navigation options: General, Access (with 'Collaborators' selected), Moderation options, Code and automation (with 'Rules', 'Actions', and 'Models' expanded), Webhooks, Copilot, Environments, Codespaces, Pages, Security (with 'Advanced Security', 'Deploy keys', and 'Secrets and variables' expanded), and Integrations (with 'GitHub Apps' and 'Email notifications'). The main content area is titled 'Collaborators and teams' and includes a 'Public repository' status box, a 'Direct access' box stating '0 collaborators have access to this repository', and a 'Manage access' section with a message 'You haven't invited any collaborators yet' and an 'Add people' button.

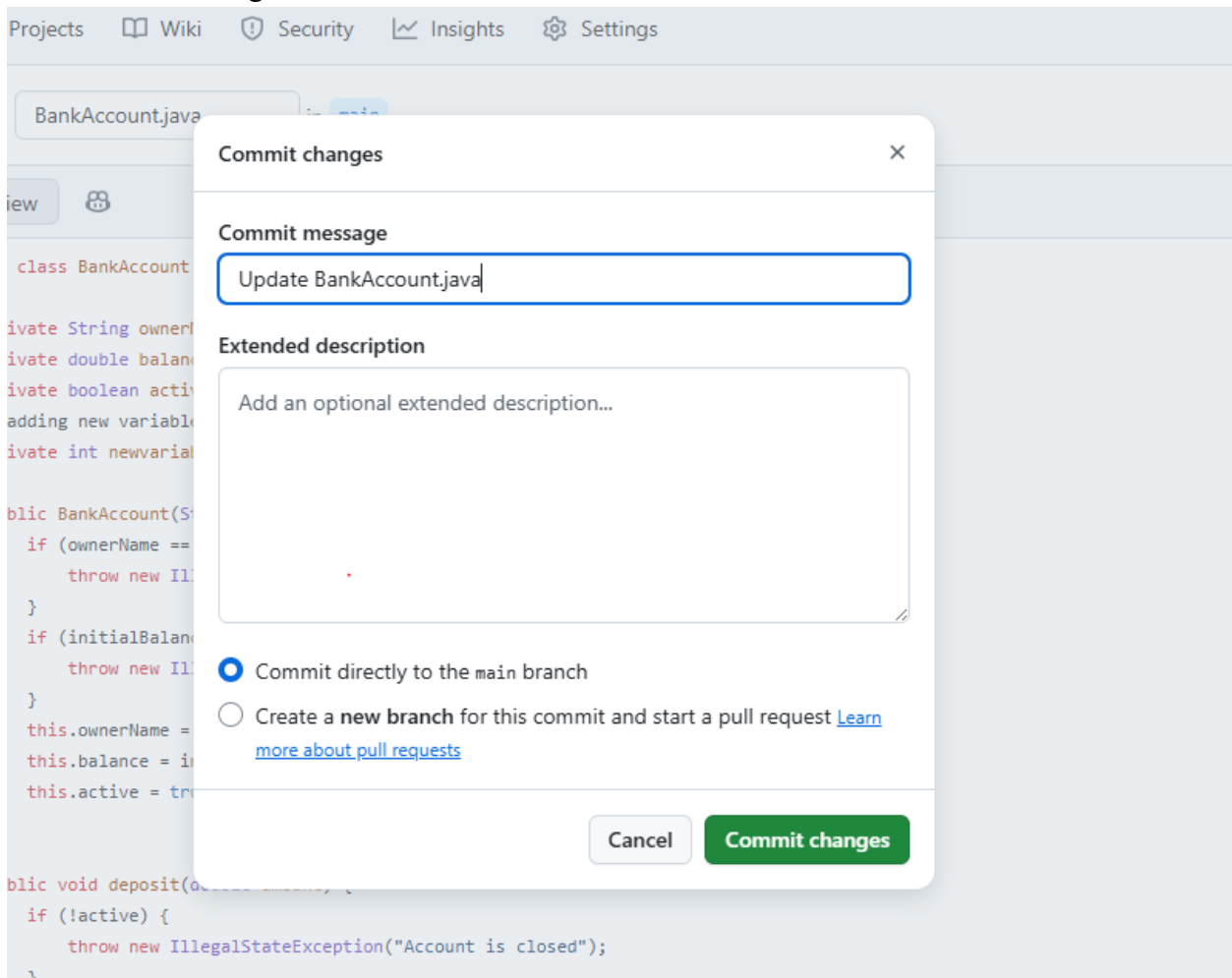
Add your code files inside the project.

The screenshot shows the main page of a GitHub repository named 'SCD-Lab-14'. The repository is public and has 1 branch and 0 tags. The file list shows a commit by 'hafuumahmood-hub' titled 'Create BankAccount.java' with a file named 'BankAccount.java'. Below the file list is a 'README' section with a message 'Add a README' and a button 'Add a README'. On the right side, there are sections for 'About' (describing the repository's purpose), 'Activity' (showing 0 stars, 0 watching, and 0 forks), 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (Java 100.0%).

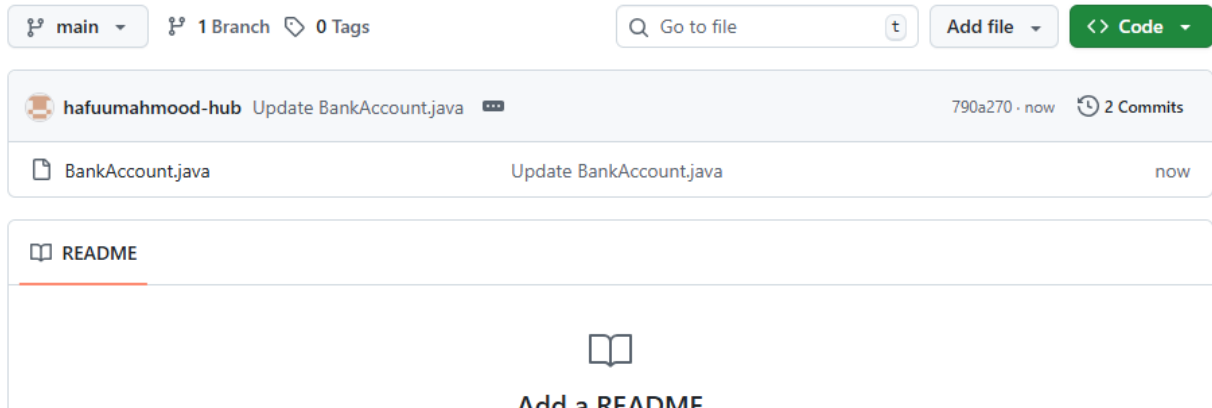
Click on commits to see the commit history



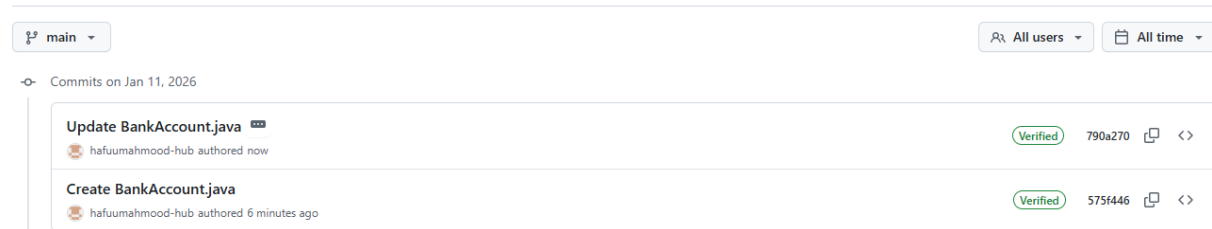
Now do some changes in the code



Now you will see two commits on the main code page



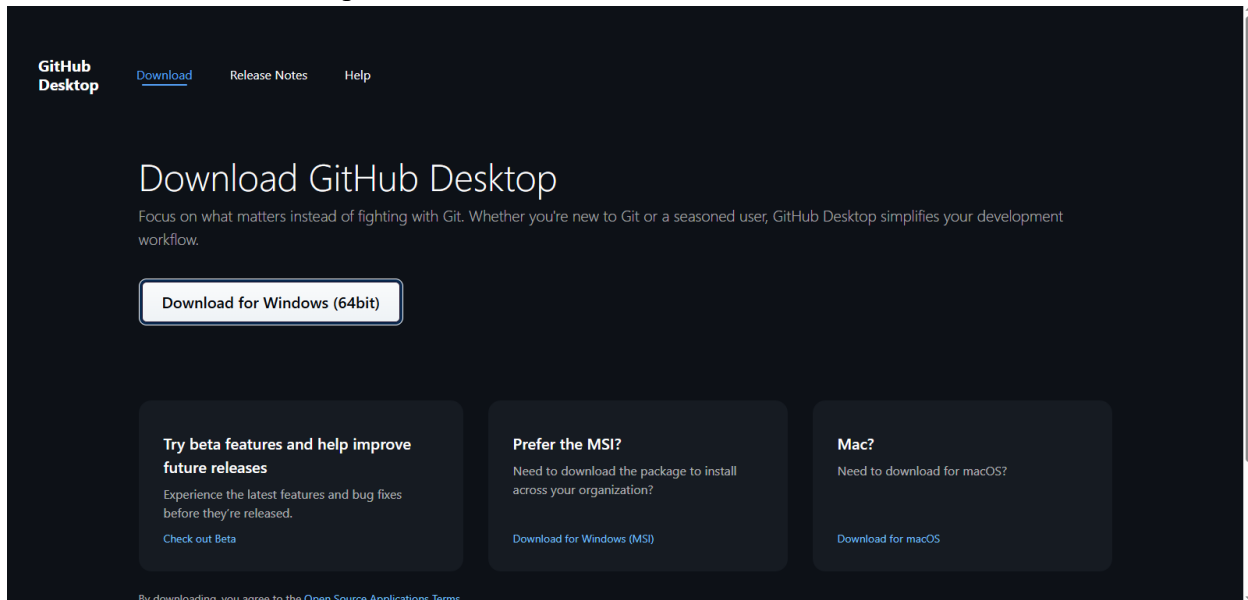
Commits



Walkthrough Task 2:

Using GitHub Desktop

Download GitHub Desktop



Clone the repository that you created in your online github account to your laptop/PC

Let's get started!

Add a repository to GitHub Desktop to start collaborating



Filter your repositories

Your repositories

hafuumahmood-hub/SCD-Lab-14

Create a tutorial repository...

Clone a repository from the Internet...

Create a New Repository on your local drive...

Add an Existing Repository from your local drive...

ProTip! You can drag & drop an existing repository folder here to add it to Desktop

Clone hafuumahmood-hub/SCD-Lab-14

Clone a repository

GitHub.com

GitHub Enterprise

URL

Repository URL or GitHub username and repository
(hubot/cool-repo)

https://github.com/hafuumahmood-hub/SCD-Lab-14.git

Local path

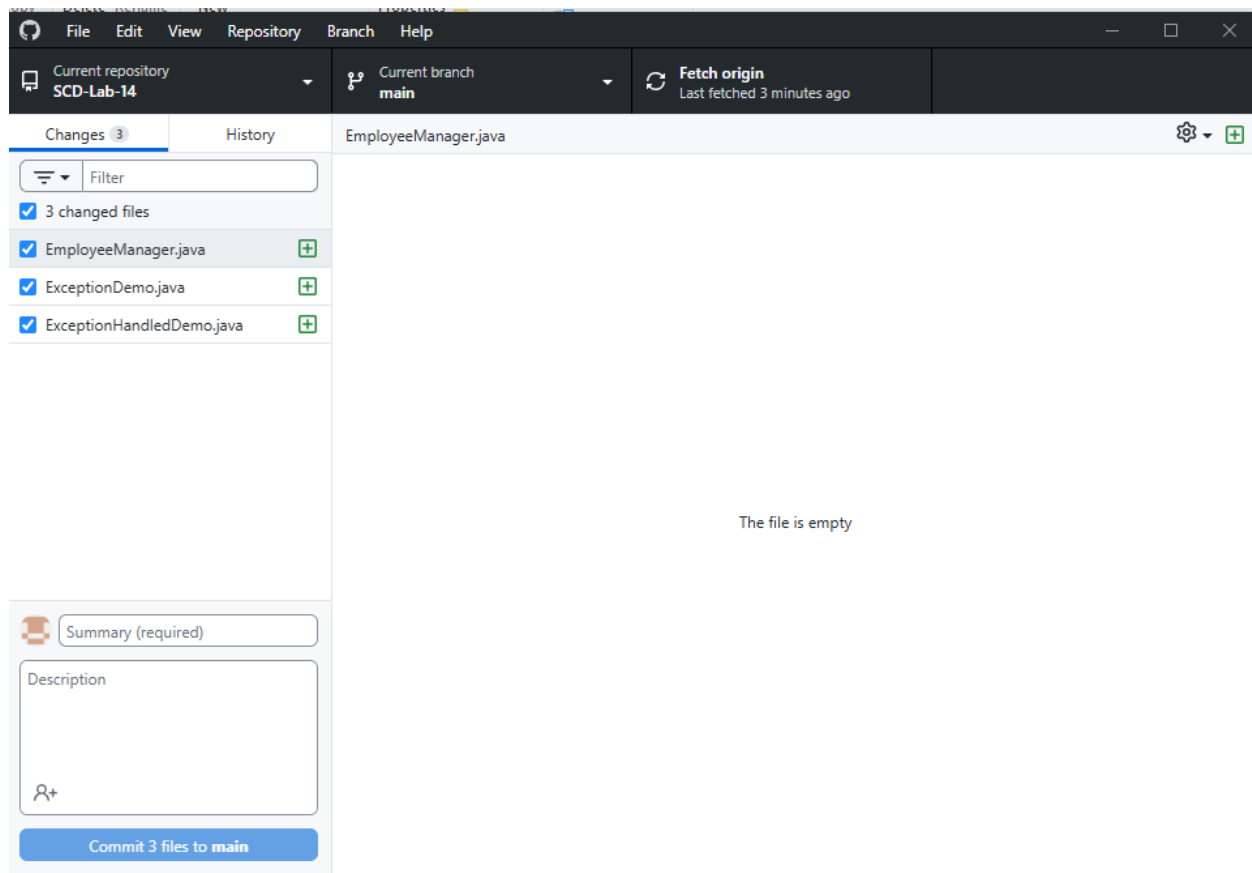
C:\Users\Hafsah\OneDrive - Punjab Group of Colleges\Documents\Gi

Choose...

Clone

Cancel

Now copy some of the files in your project folder



Now commit these new added files to main

File Edit View Repository Branch Help

Current repository: SCD-Lab-14

Current branch: main

Push origin (1 ↑)
Last fetched 4 minu...

Changes History

Filter

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl + P**

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or **Ctrl + Shift + A**

View the files of your repository in Explorer
Repository menu or **Ctrl + Shift + F**

Open the repository page on GitHub in your browser
Repository menu or **Ctrl + Shift + G**

Summary (required)

Description

Commit to main

Committed just now
Three new files are added

Push the commits to origin remote

After that your original repository on the internet will also reflect the changes that you made inside your system.

SCD-Lab-14 (Public)

Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

hafuumahmood-hub Three new files are added 12cf35f · 1 minute ago 3 Commits

File	Commit Message	Time
BankAccount.java	Update BankAccount.java	36 minutes ago
EmployeeManager.java	Three new files are added	1 minute ago
ExceptionDemo.java	Three new files are added	1 minute ago
ExceptionHandledDemo.java	Three new files are added	1 minute ago

README

About
This will cover the basics of how to create the repository and commit changes in it

Activity
0 stars
0 watching
0 forks

Releases
No releases published
[Create a new release](#)

Commits

main

All users All time

Commits on Jan 11, 2026

Three new files are added
hafuumahmood-hub committed 2 minutes ago
12cf35f

Update BankAccount.java
hafuumahmood-hub authored 37 minutes ago
Verified 790a270

Create BankAccount.java
hafuumahmood-hub authored 42 minutes ago
Verified 575f446

Now if you make any changes on the script on your web GitHub you will need to pull those changes into your desktop app.

File Edit View Repository Branch Help

Current repository SCD-Lab-14 Current branch main Pull origin Last fetched just now 1

Changes History

Filter

0 changed files

Summary (required)

Description

Commit to main

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull 1 commit from the origin remote
The current branch (main) has a commit on GitHub that does not exist on your machine.
Always available in the toolbar when there are remote changes or `Ctrl + Shift + P`

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or `Ctrl + Shift + A`

View the files of your repository in Explorer
Repository menu or `Ctrl + Shift + F`

Open the repository page on GitHub in your browser
Repository menu or `Ctrl + Shift + G`

Click on pull origin and your both directories will be synced.

Practice Tasks:

Practice Task 1: Team-Based Git Collaboration on a Mini Application

Objective

To enable students to collaboratively build different modules of a small management system while using GitHub as the shared version control platform for integration, conflict handling, and synchronization

Task Description

Students are required to perform the following steps:

1. **Form teams of 3–4 members.**
2. **Select one mini application to develop**, such as (examples):
 - Library Management System
 - Inventory Management System
 - Course Registration System
 - Hostel/Room Allocation System
 - Clinic Appointment System
 - Vehicle Service Tracking System
 - Simple Payroll System
3. **Assign modules to each member**, such as:
 - Member A → User/Student/Employee Module
 - Member B → Resource/Inventory/Asset Module
 - Member C → Records/Transactions/Bookings
 - Member D (optional) → Reporting / Authentication / Admin Panel
4. **Create a single GitHub repository** for the team and invite all members as collaborators.
5. Each member must:
 - Clone the repository
 - Develop their assigned module locally
 - Commit changes with meaningful messages
 - Push changes to the main project
6. Teams must create **at least one separate branch per member**, such as:
 - a) module-user
 - b) module-inventory
 - c) module-transactions
 - d) module-report

7. System must include **at minimum**:
 - Basic CRUD operations for each module
 - Input validation & error handling
 - Simple console or GUI interface
8. Final submission artifacts:
 - Repository link
 - Screenshots of commits
 - Screenshot of merges

Outcomes:

- Students will understand version control concepts using Git.
- Students will be able to stage, commit, push, and pull code using GitHub.
- Students will handle real-world synchronization issues and merge conflicts.
- Students will collaborate through GitHub repositories.
- Students will appreciate GitHub as a professional development, deployment, and teamwork tool.