University of Houston-Victoria

Department of Computer and Information Science

Group Project

Expense Tracker

Deployment Manual

COSC 6342 Software Engineering Project Management

Spring 2024

By

Aradhana Sharma (2250955)

Kapil Suryawanshi (2039409)

Shayan Khan (2244554)

Suraj Odedra (2329291)

Advisor: Dr. Amjad Nusayr

Director of UG Computer Science & Computer Information Systems  
Associate Professor of Computer Science

University of Houston Victoria

Table of Contents

[Abstract 4](#_Toc158808060)

[Development Environmet 4](#_Toc158808061)

[Visual Studio Code 4](#_Toc158808062)

[GitHub 9](#_Toc158808063)

[Git 15](#_Toc158808064)

[Node.js 27](#_Toc158808065)

[MySQL 33](#_Toc158808066)

[Install Postman 46](#_Toc158808067)

[Get all users 49](#_Toc158808068)

[Get a user by id 49](#_Toc158808069)

[Create user 50](#_Toc158808070)

[Update user 50](#_Toc158808071)

[Delete user 51](#_Toc158808072)

[Setup Git <->GitHub communication 51](#_Toc158808073)

[Setup Front End app 61](#_Toc158808074)

[Setup Back End App 63](#_Toc158808075)

[Production Environment 65](#_Toc158808076)

[Archived 65](#_Toc158808077)

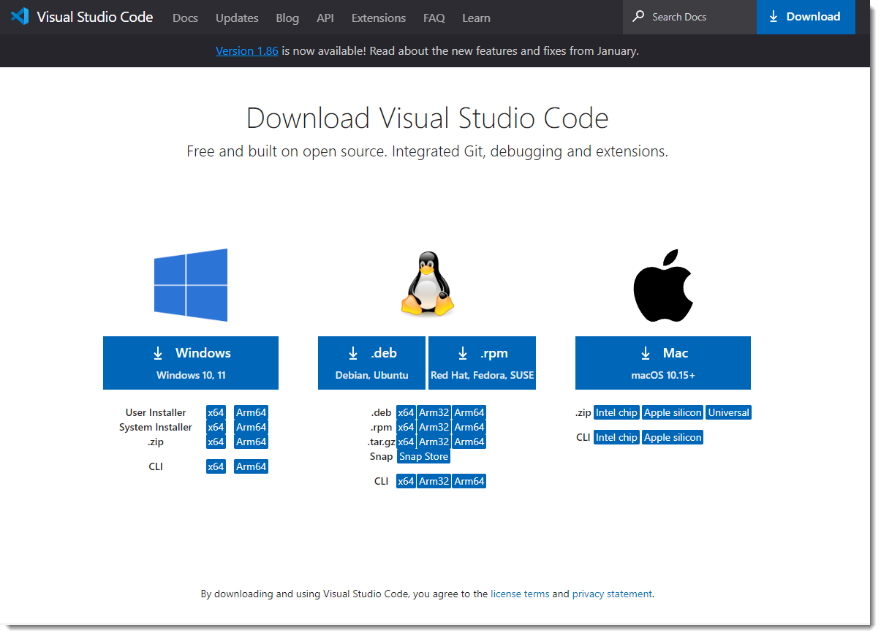
# Abstract

The document is intended to describe the setup of the software tools required for preparing the development environment. It also describes the deployment protocol to be followed to deploy the developed software.

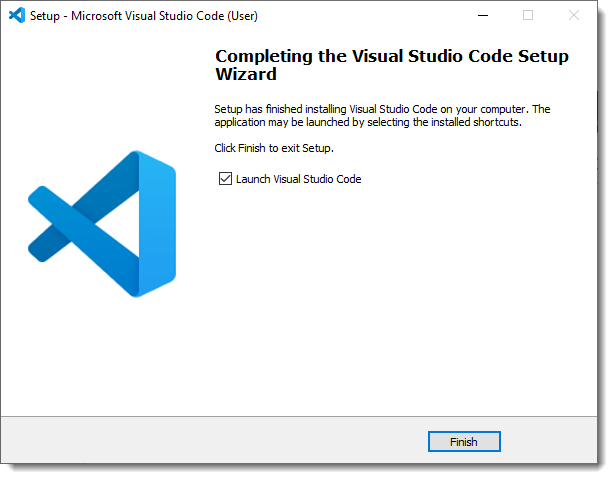
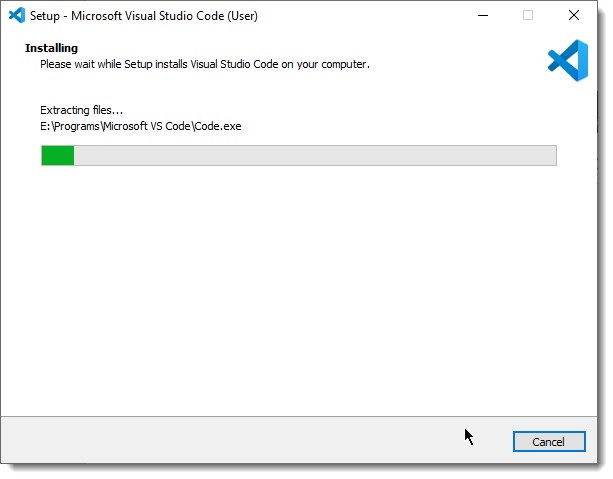
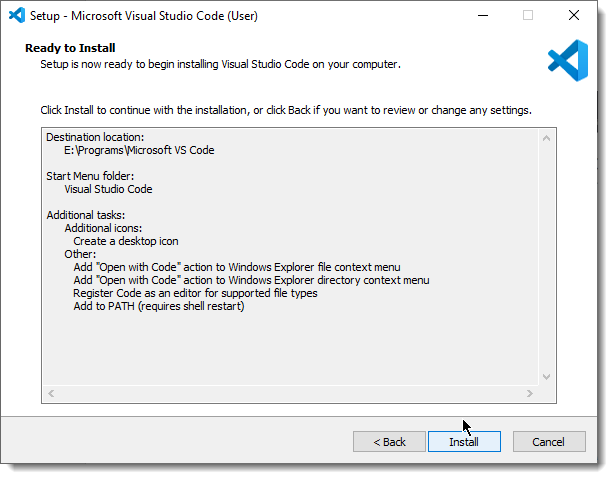
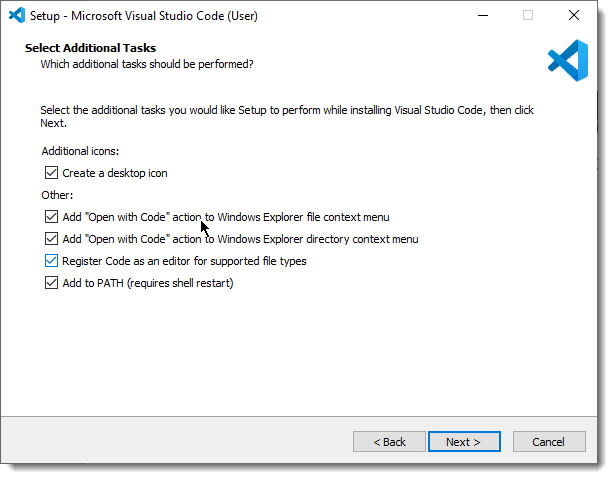
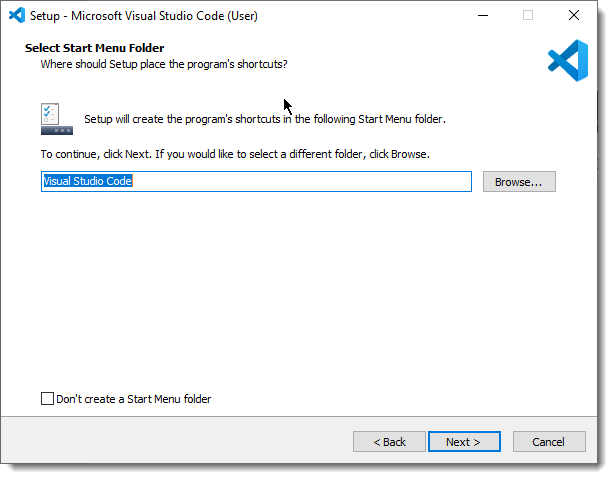
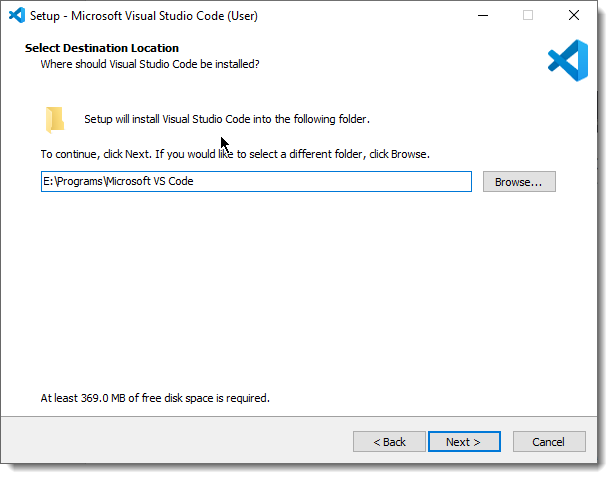
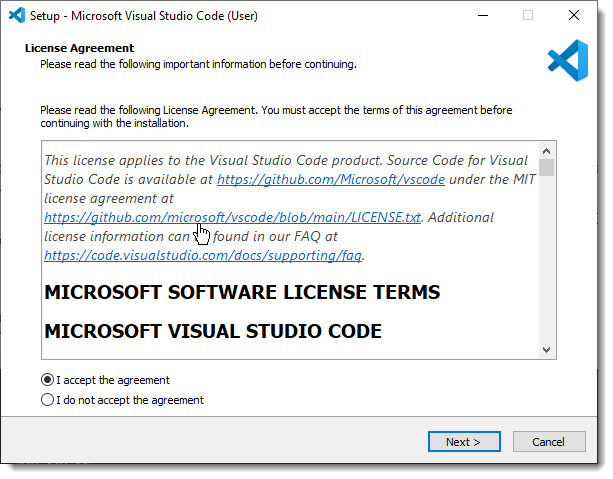
# Development Environmet

## Visual Studio Code

Navigate to <https://code.visualstudio.com/download>

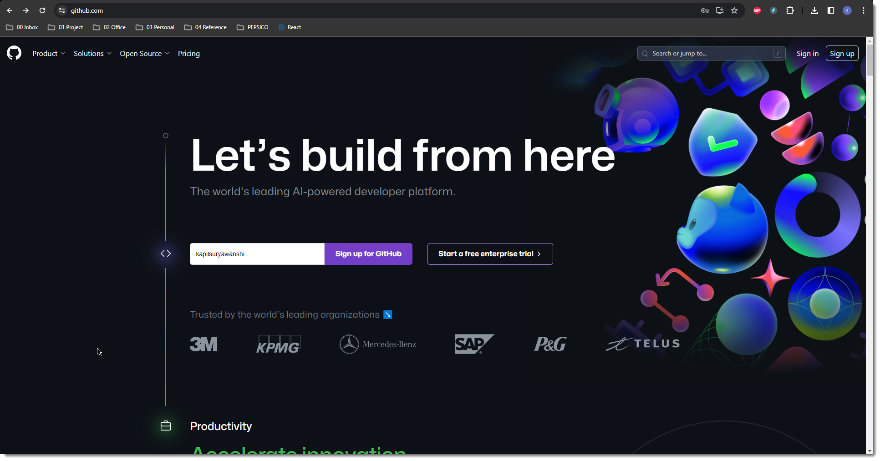


Download the required executable as per your operating system. Here I am downloading installer for Windows OS. Once downloaded double click the executable and start the installer. Following screens shows the options selected during a typical installation.

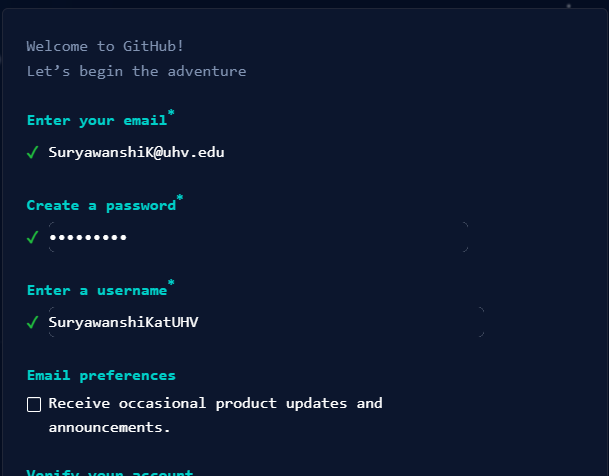


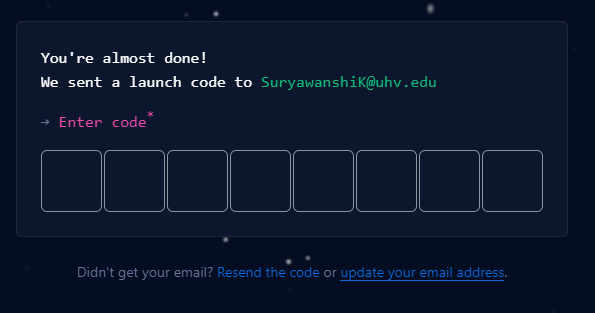
## GitHub

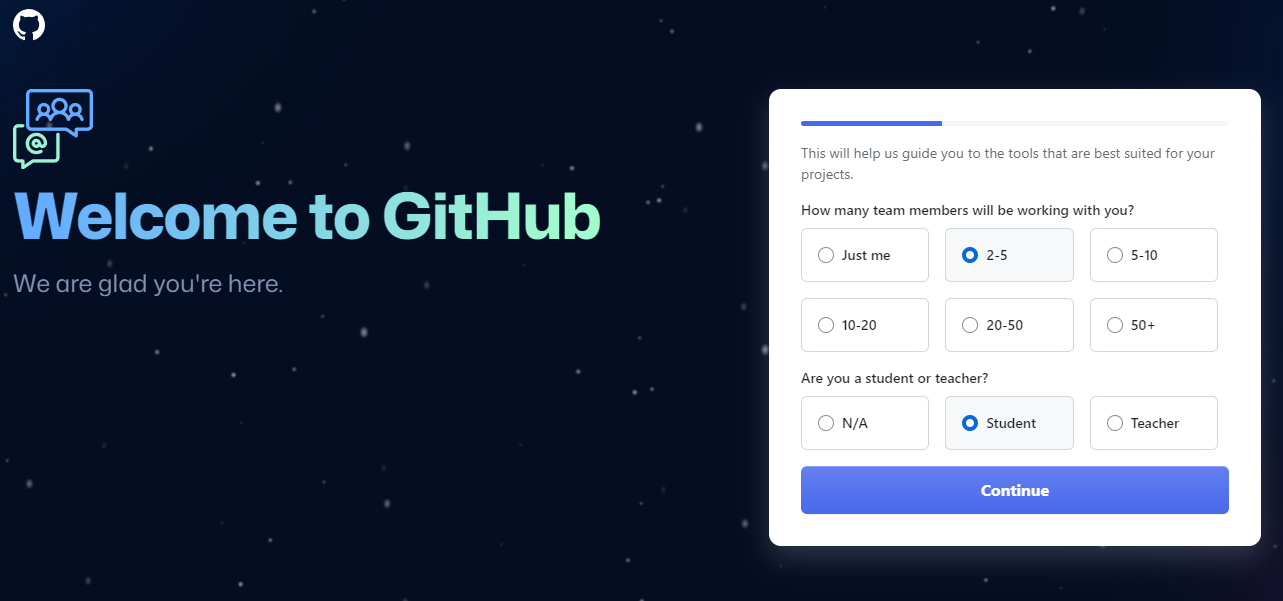
Navigate to <https://github.com/>

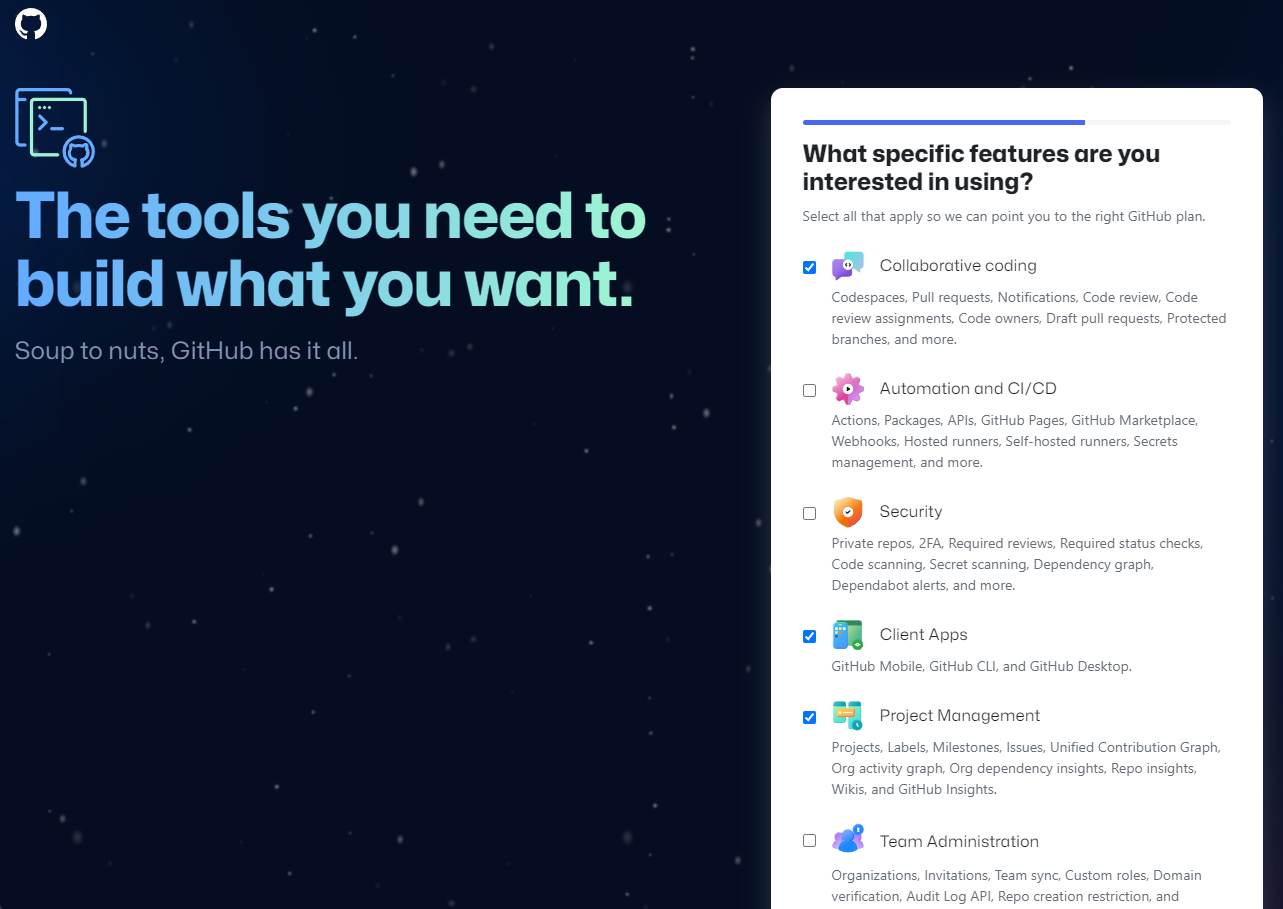


Click Sign Up. Create username and password.





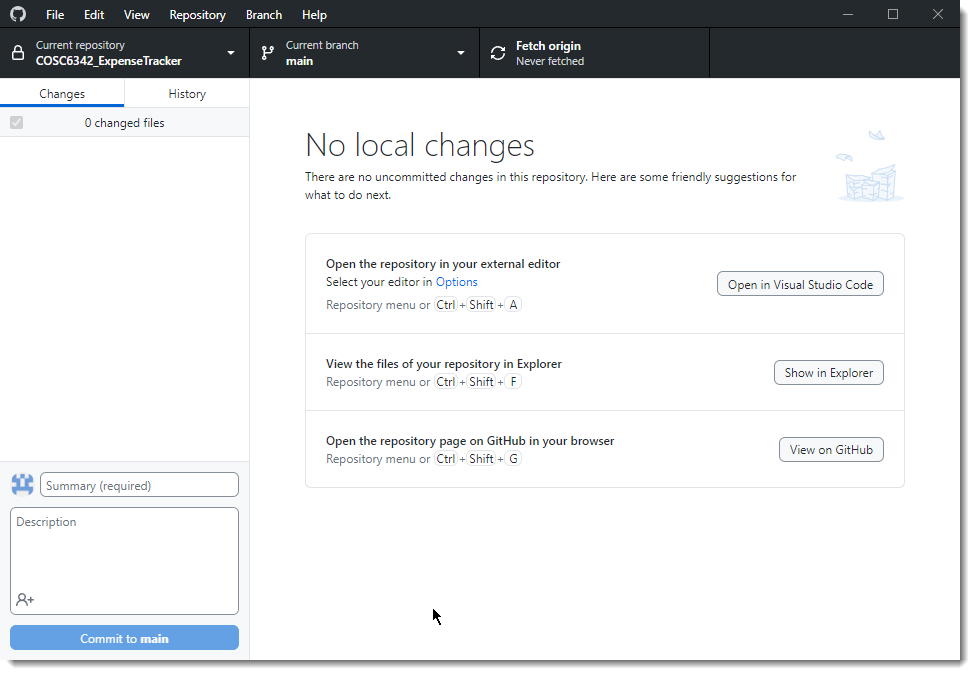
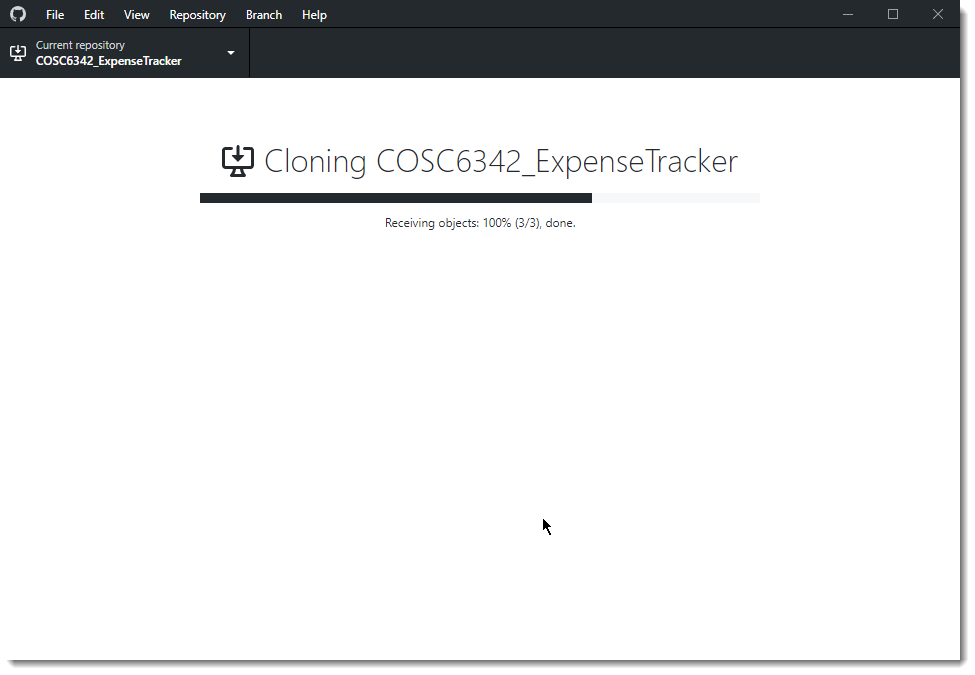
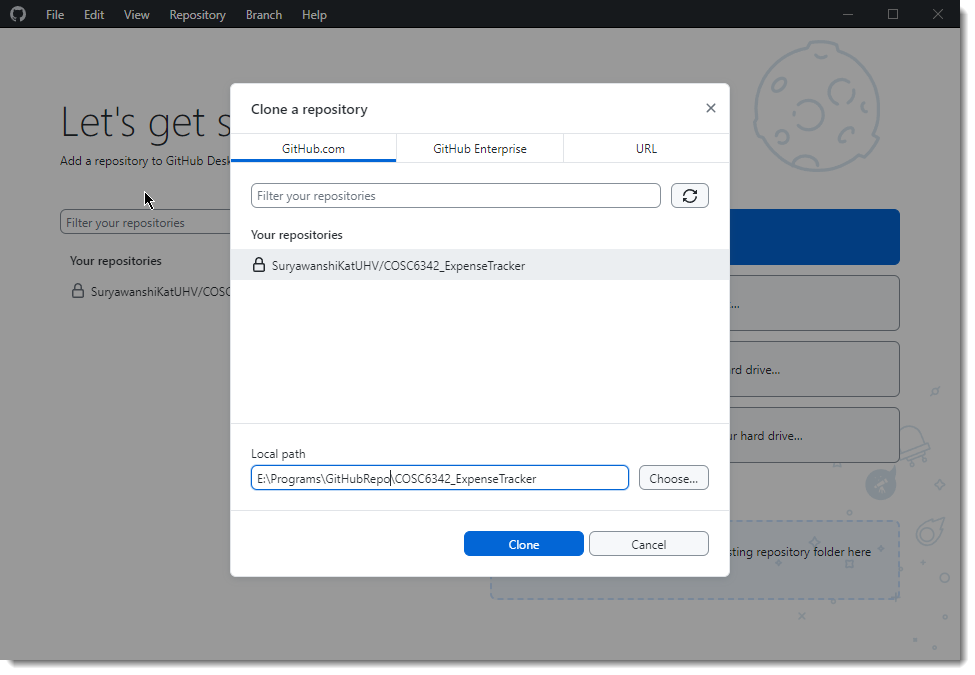
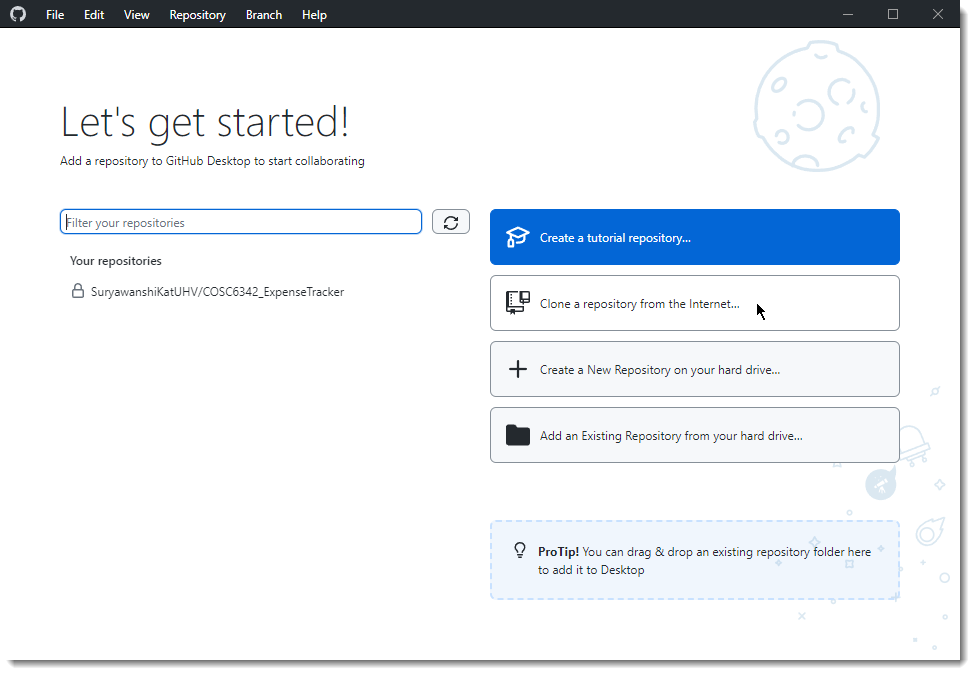
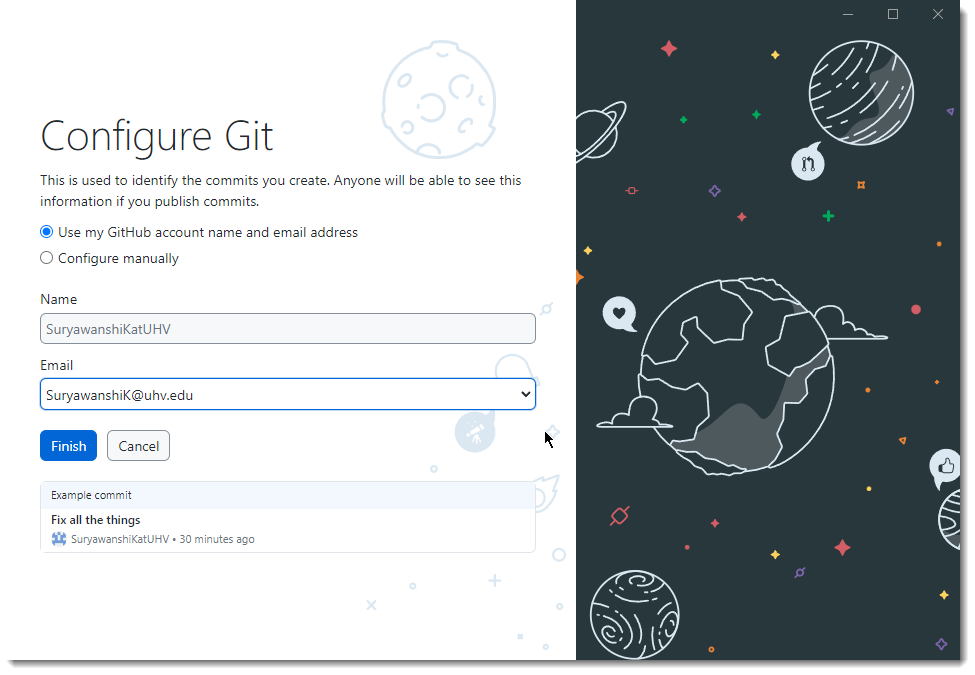
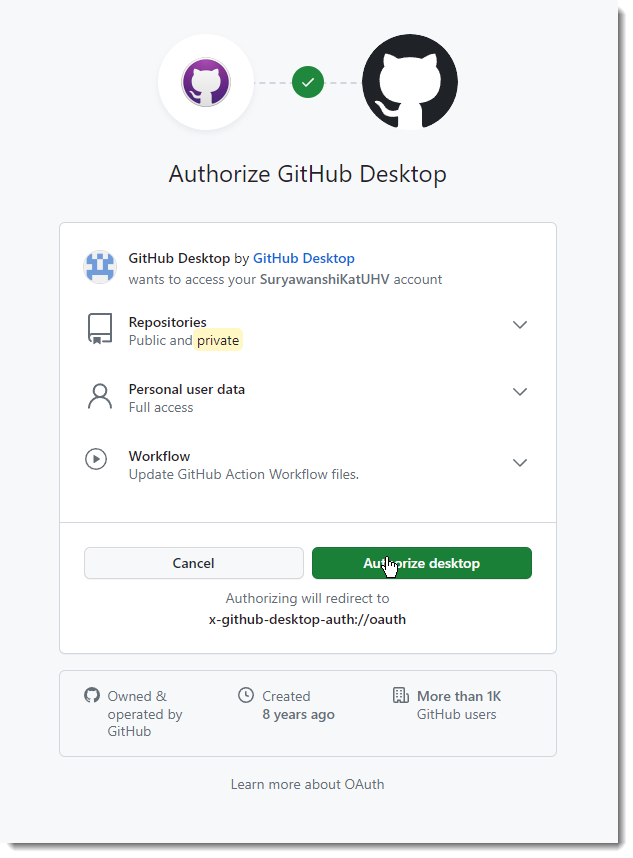
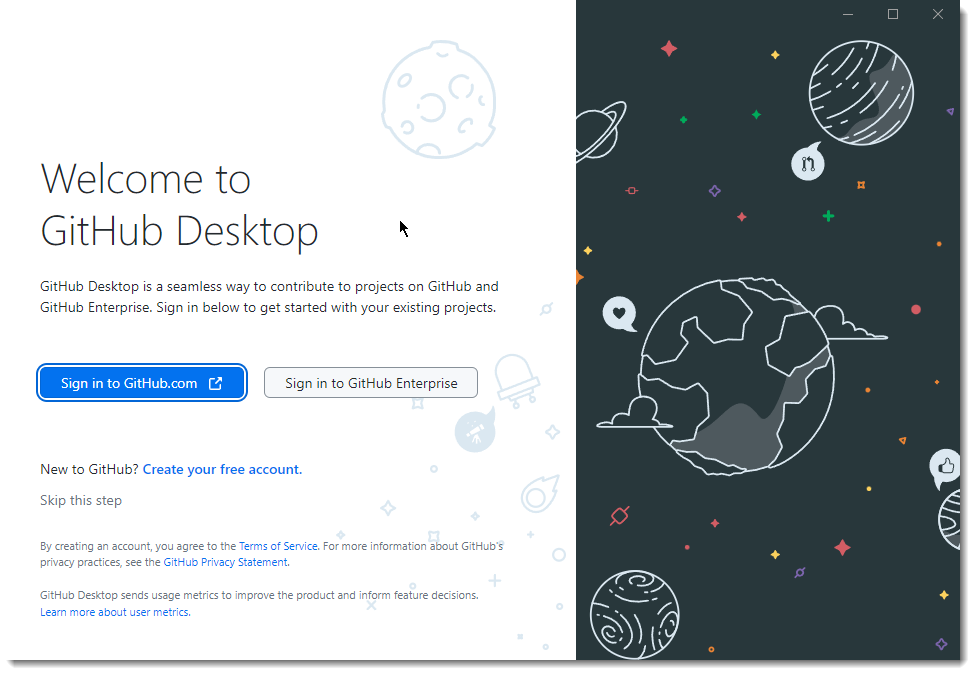




I have created a repository COSC6342\_ExpenseTracker

(Optional) Now download GitHub Desktop from <https://desktop.github.com/>

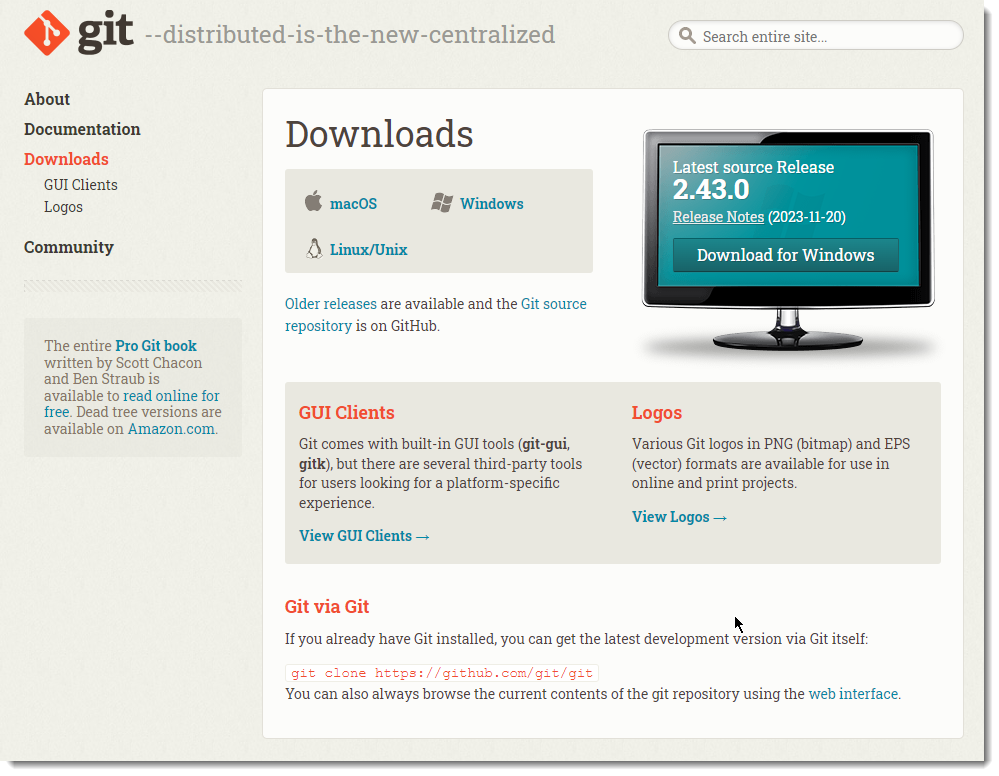




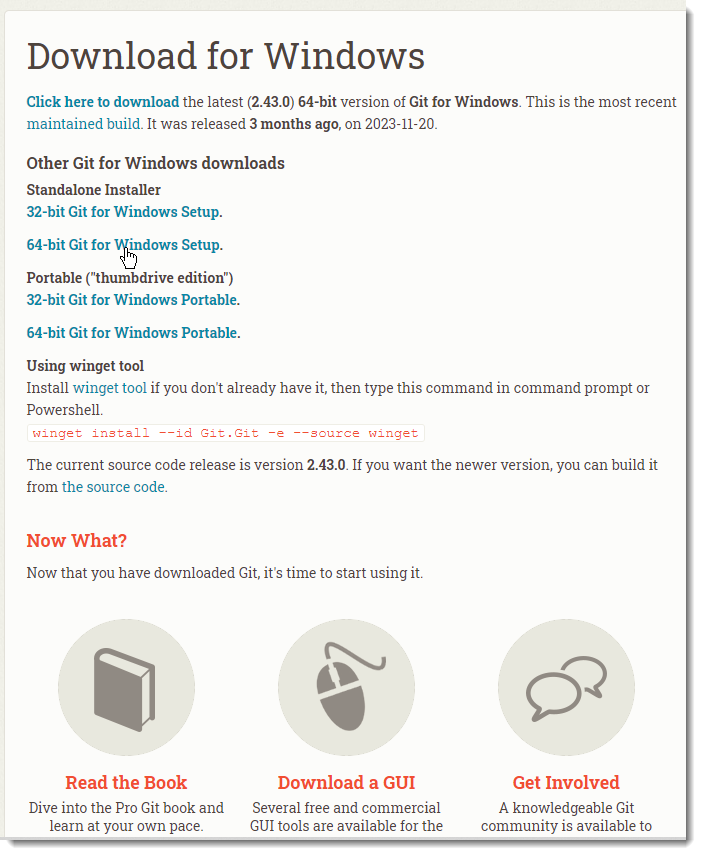
End

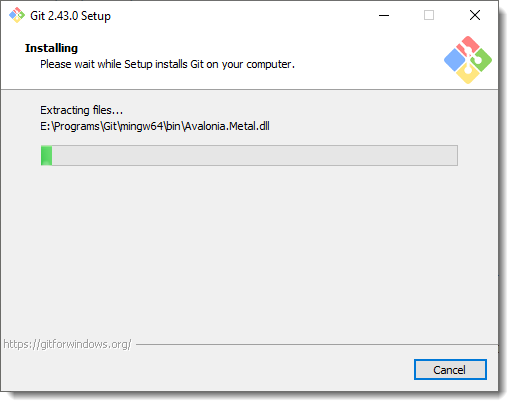
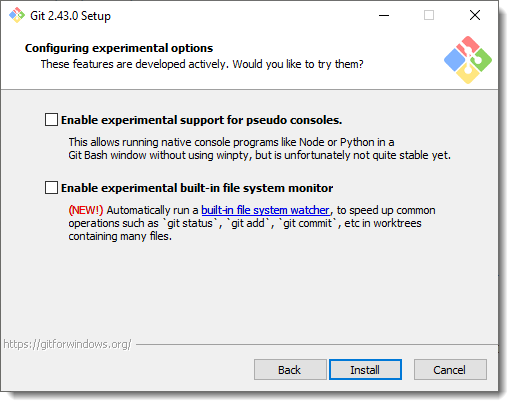
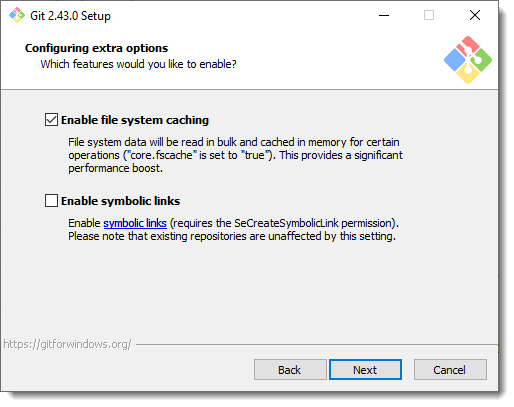
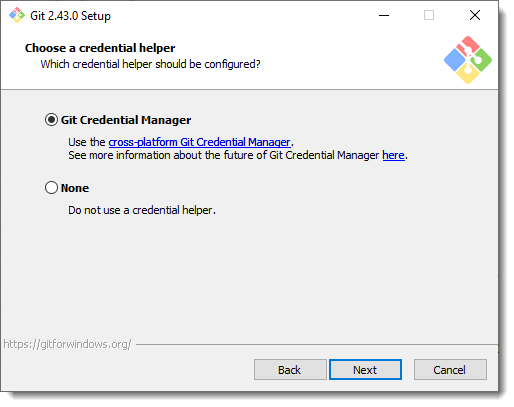
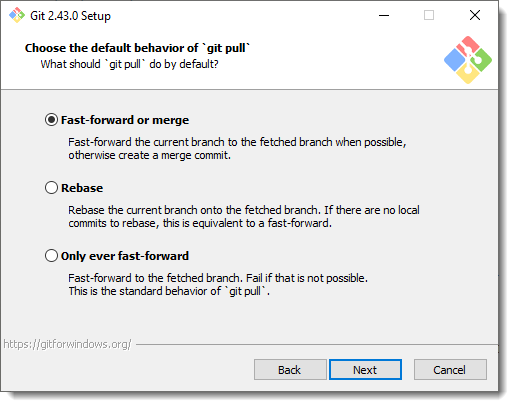
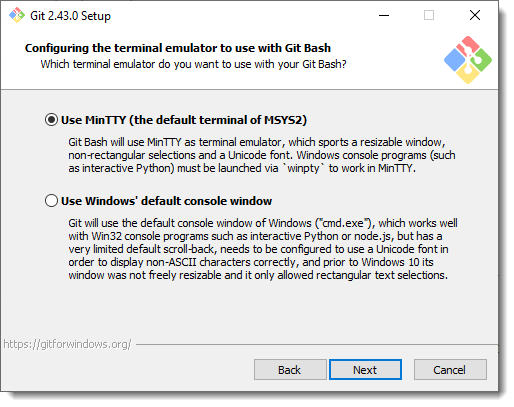
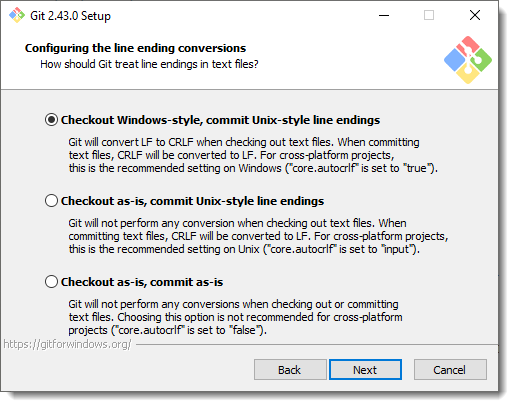
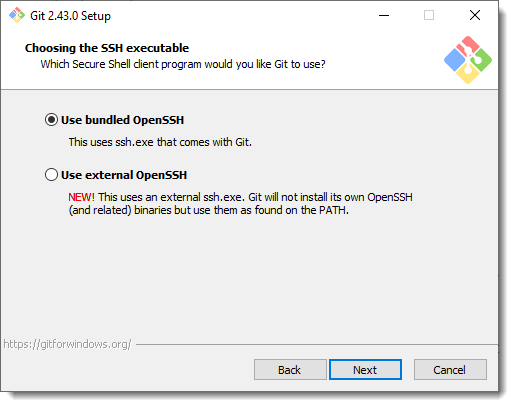
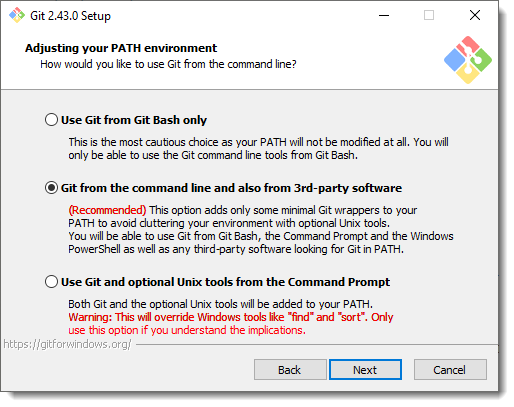
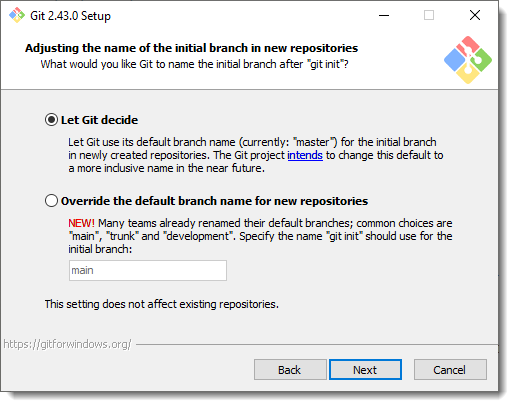
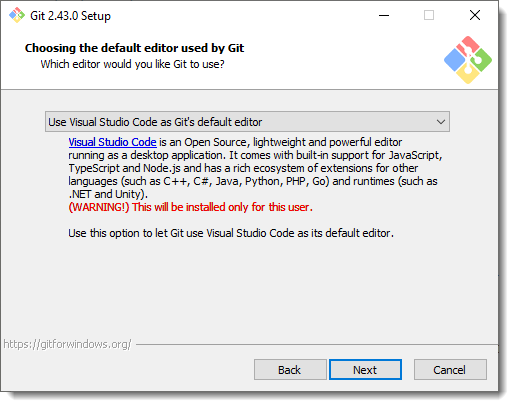
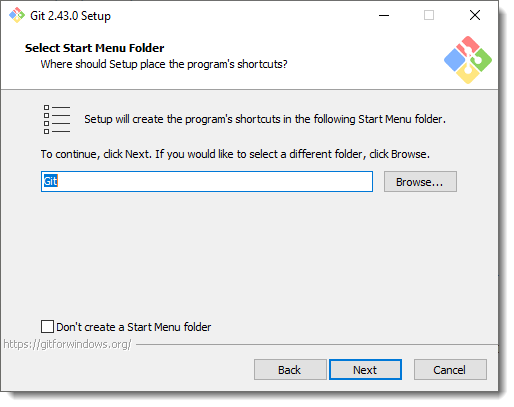
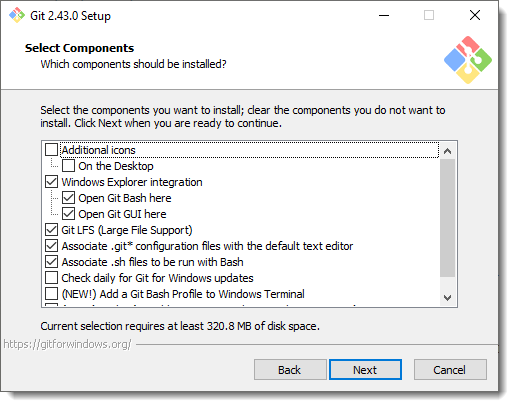
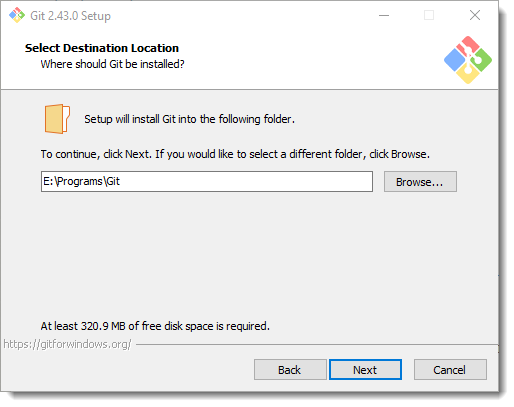
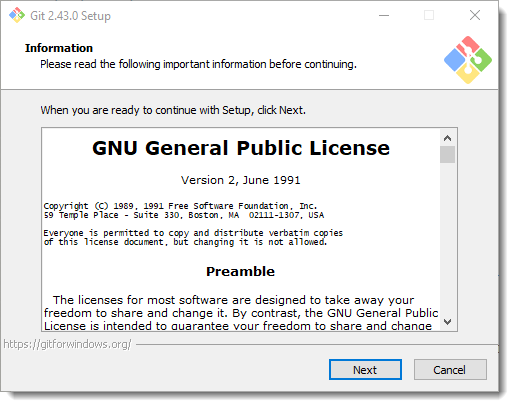
## Git

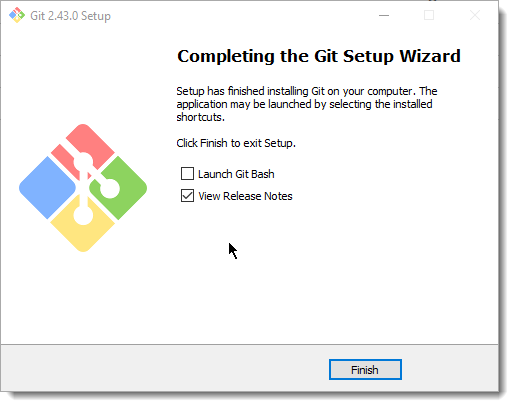
Navigate to <https://git-scm.com/downloads>



Download the required executable as per your operating system. Here I am downloading installer for Windows OS. Once downloaded double click the executable and start the installer. Following screens shows the options selected during a typical installation.







There are two ways to configure your username and email in Git settings: globally for all repositories on your computer, or locally for specific repositories. Here's how to do it for each:

* **Globally**:

1. Open your terminal.
2. Set your username: Use the command git config --global user.name "YOUR\_USERNAME". Replace YOUR\_USERNAME with your preferred username.
3. Set your email: Use the command git config --global user.email "YOUR\_EMAIL". Replace YOUR\_EMAIL with your actual email address.
4. Verify your settings: Use the command git config --global --list to check if your username and email are set correctly.



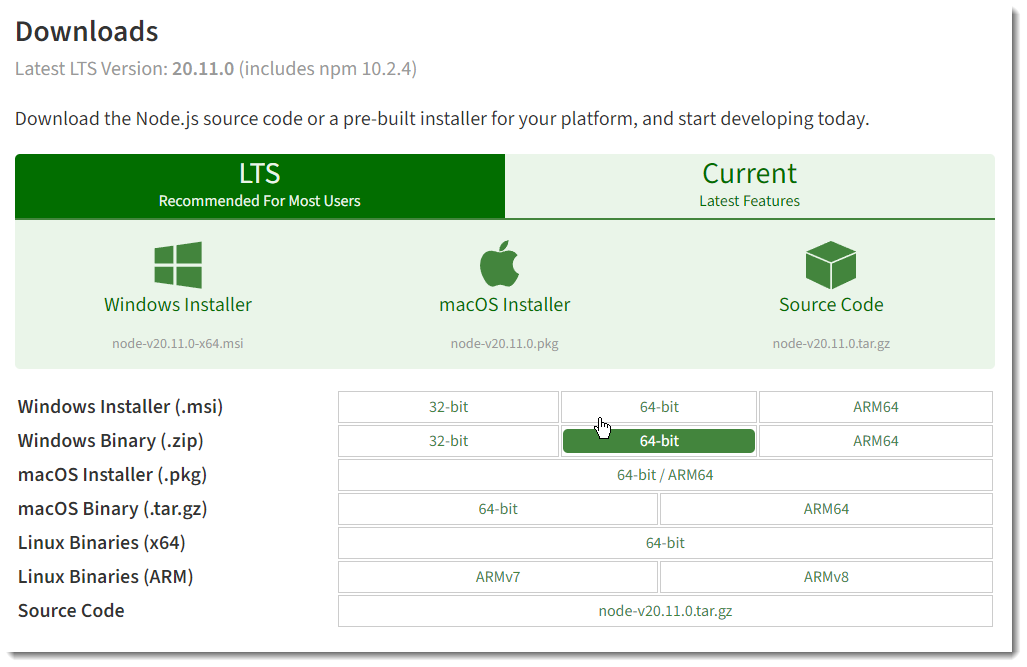
* **Locally:**

1. Navigate to the desired repository directory.
2. Set your username: Use the command git config user.name "YOUR\_USERNAME".
3. Set your email: Use the command git config user.email "YOUR\_EMAIL".
4. Verify your settings: Run git config --list to see the local configuration for that repository.

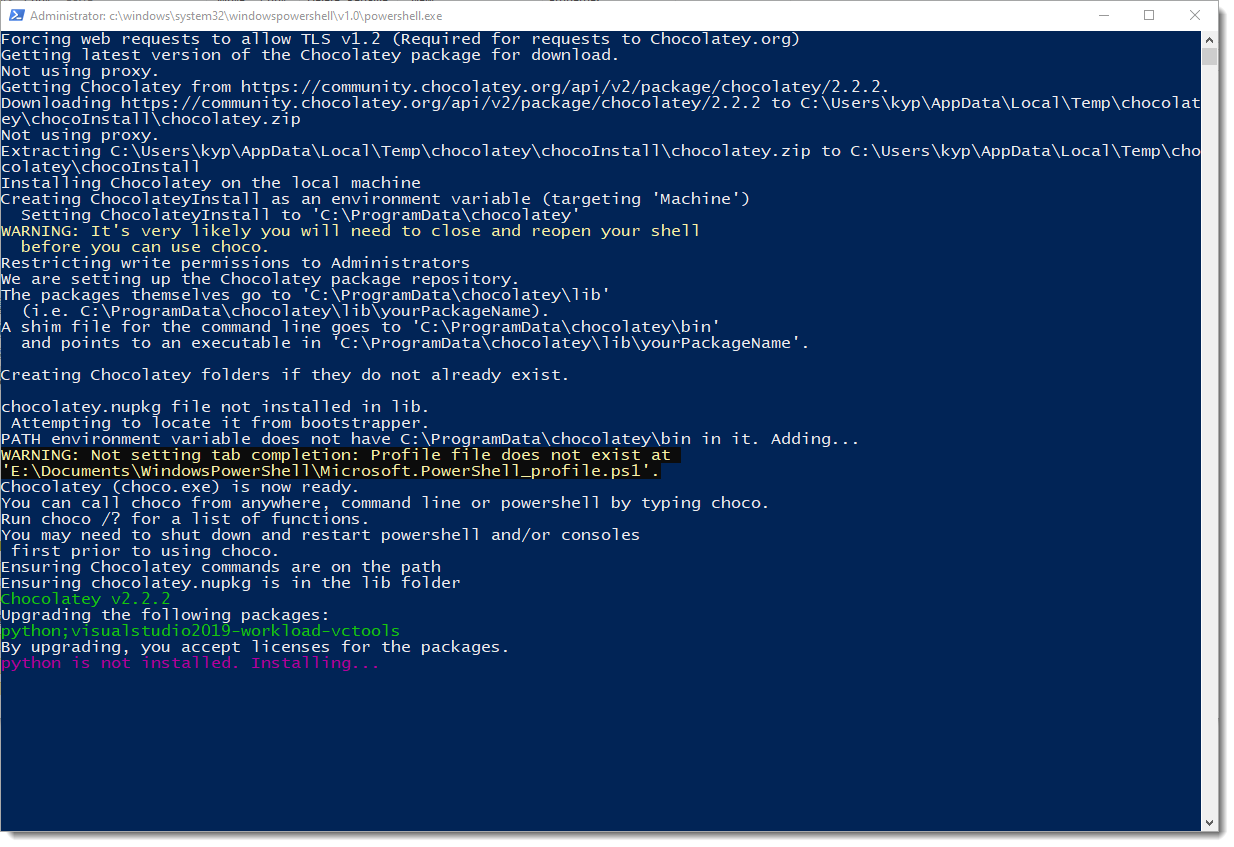
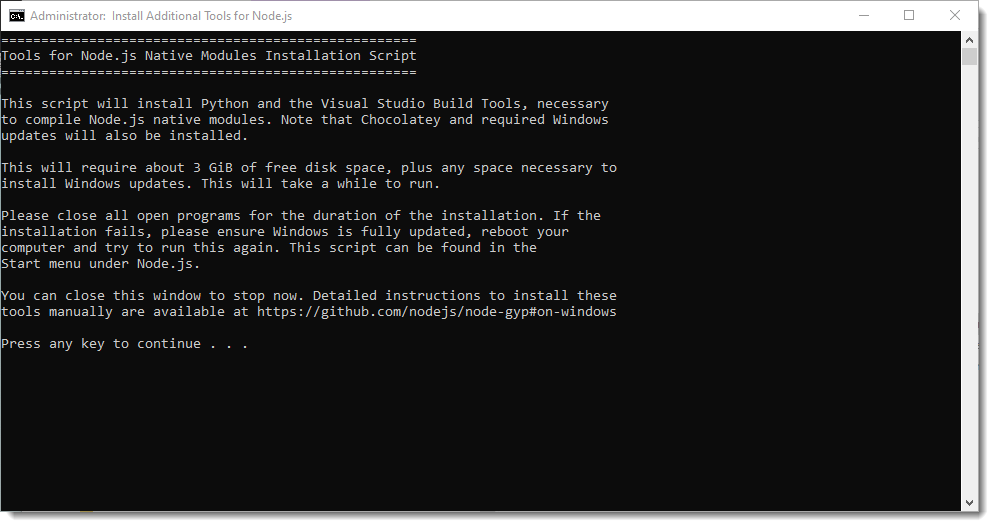
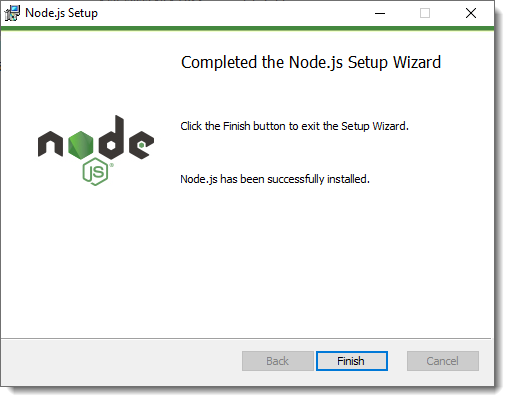
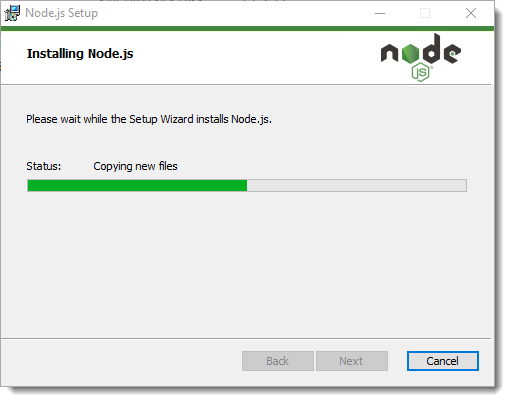
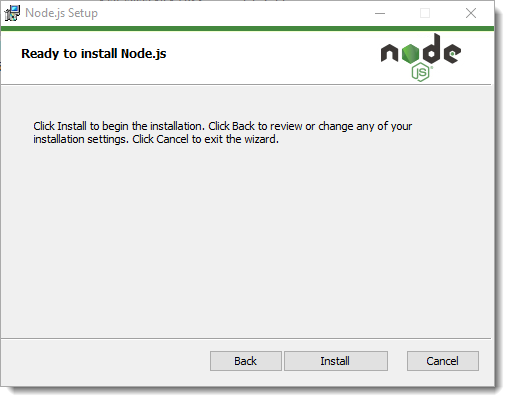
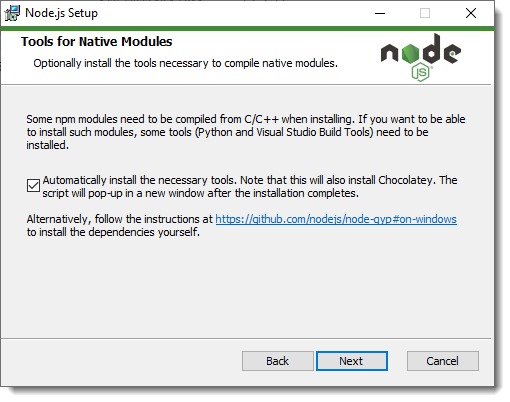
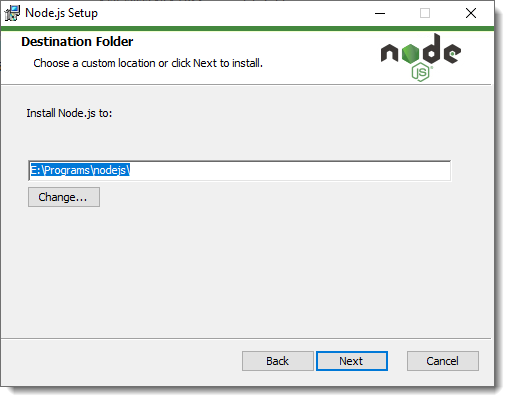
* **Additional tips:**
* You can use quotes around your username and email if they contain spaces.
* Consider using a separate email address for Git commits if you want to keep your primary email private.
* Some platforms like GitHub may have additional settings for configuring your username and email.

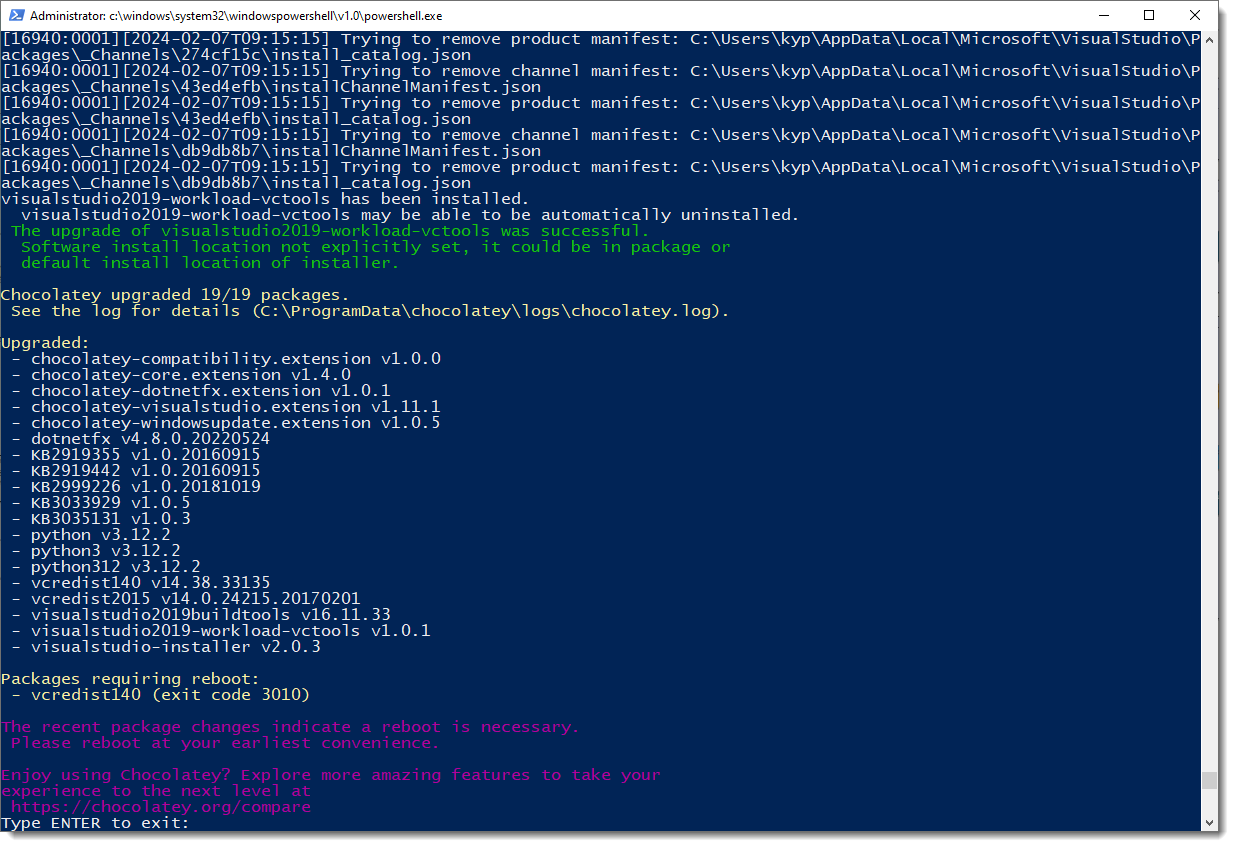
## Node.js

Navigate to <https://nodejs.org/en/download>



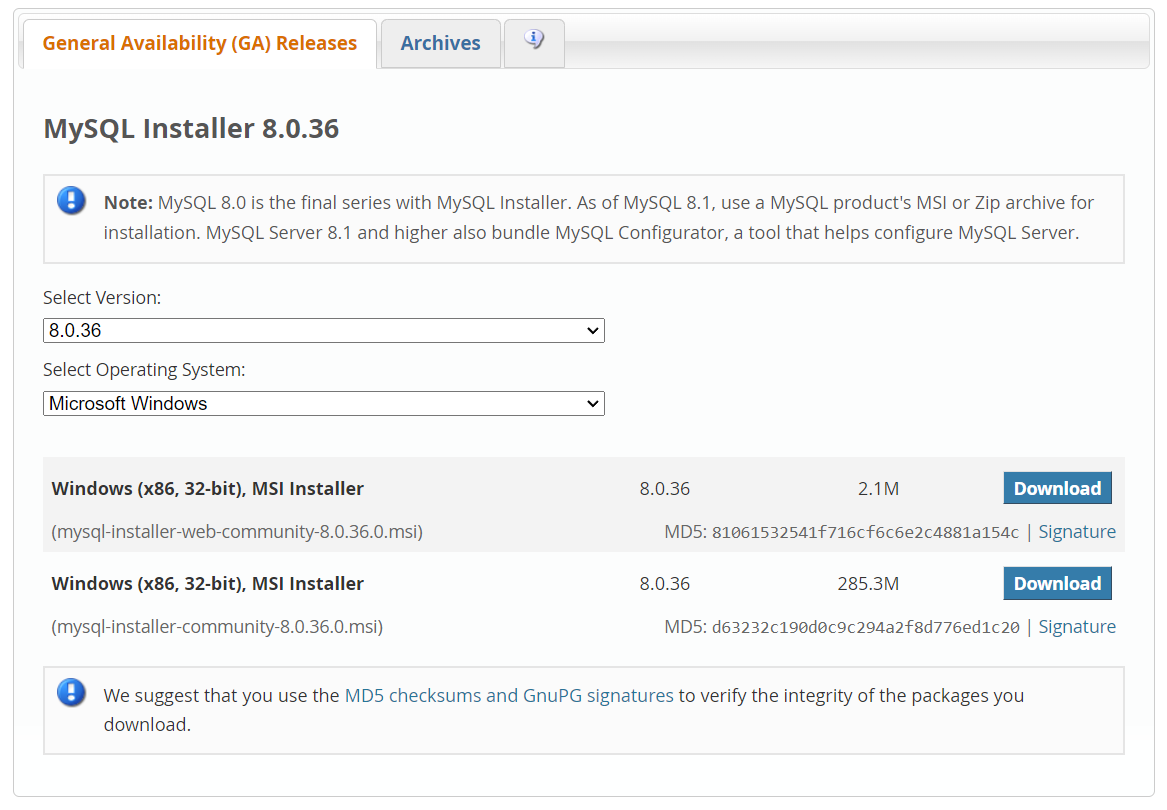
Download the required executable as per your operating system. Here I am downloading installer for Windows OS. Once downloaded double click the executable and start the installer. Following screens shows the options selected during a typical installation.

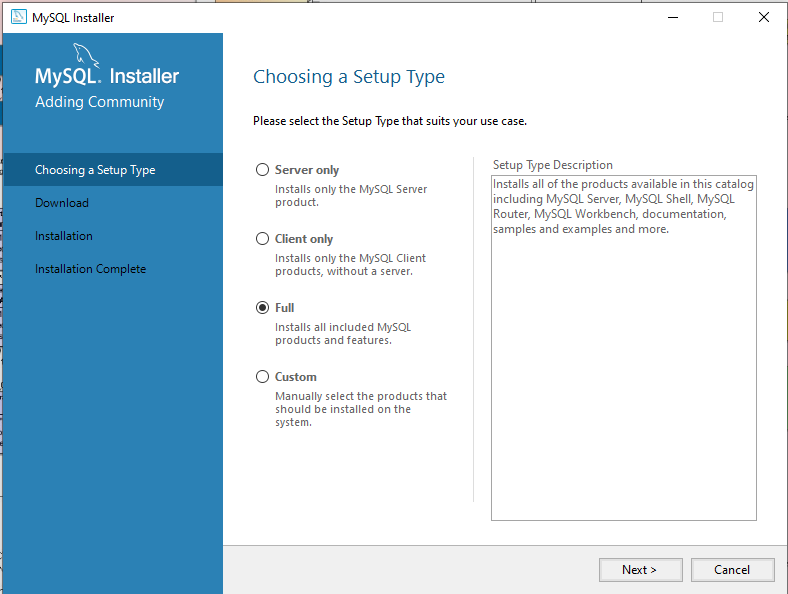


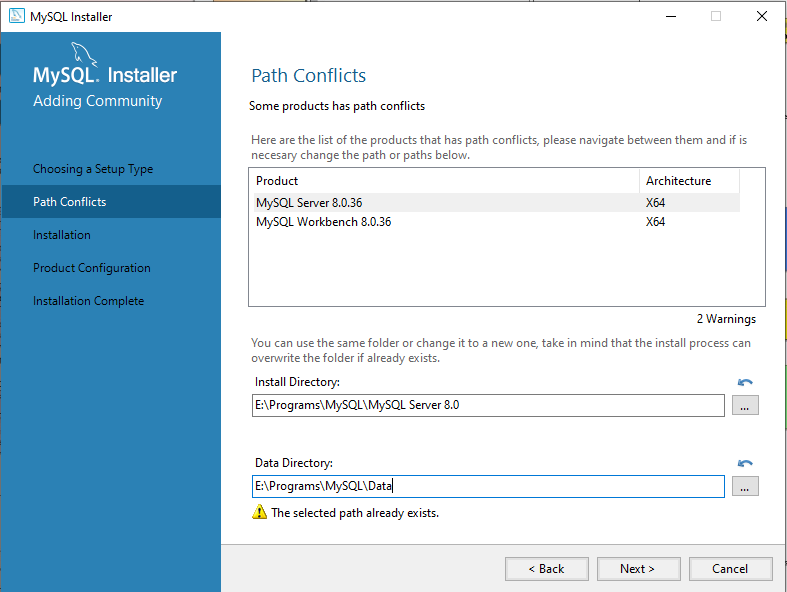


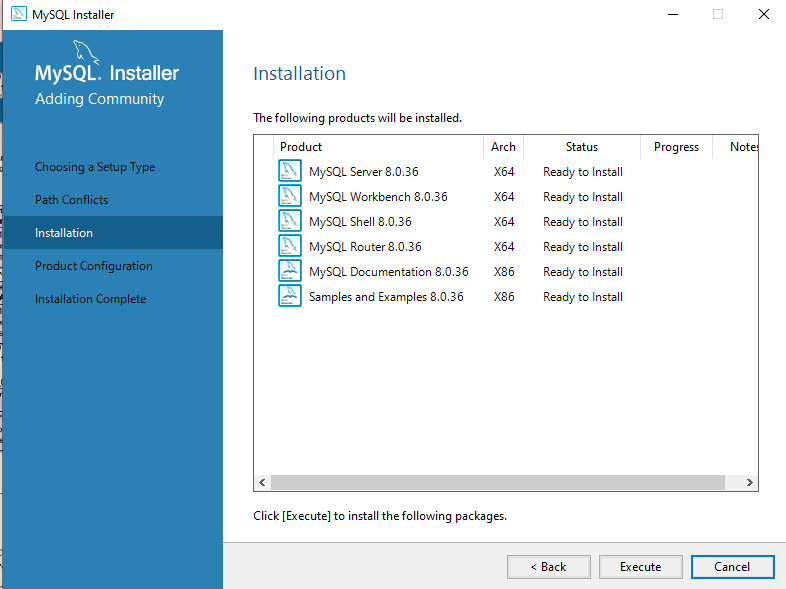
## MySQL

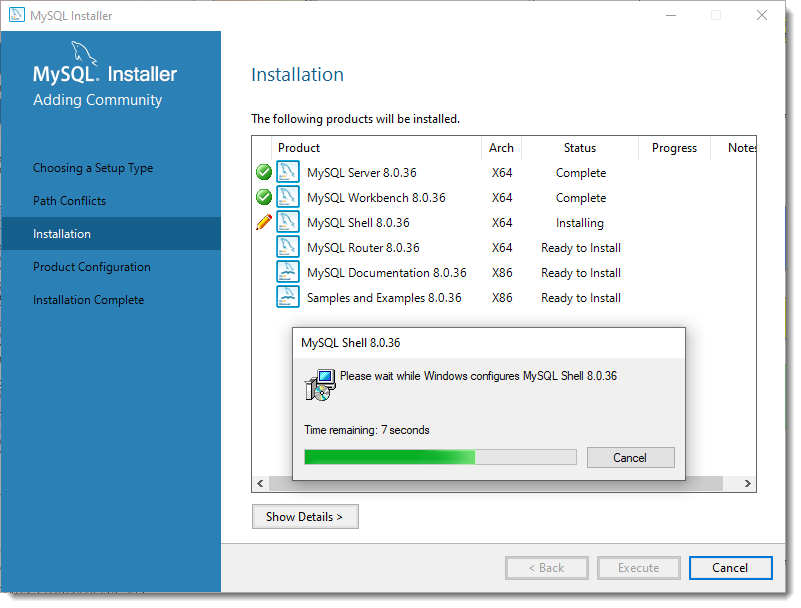
Navigate to <https://dev.mysql.com/downloads/installer/> and download the installer for your platform.

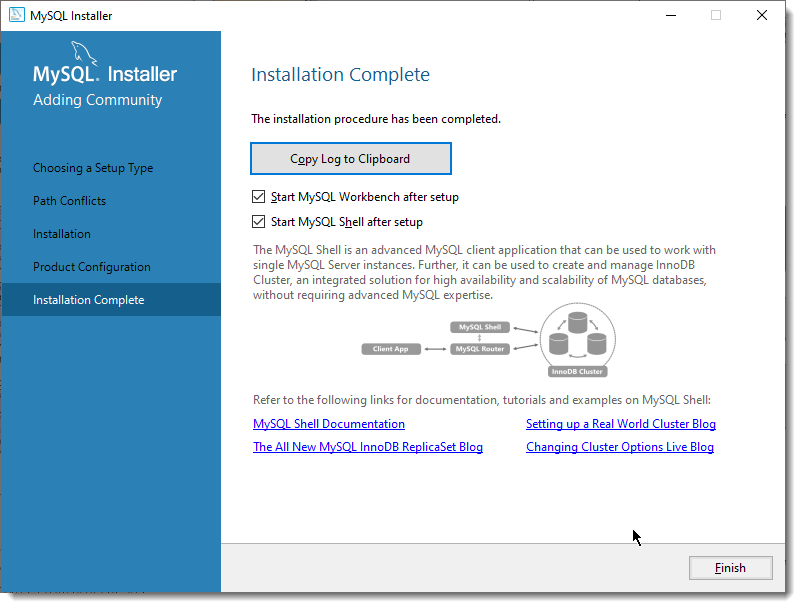
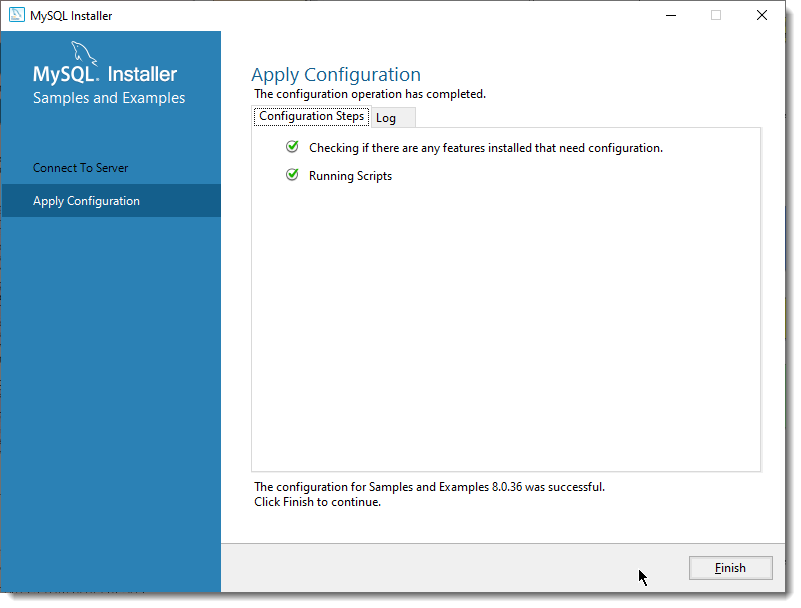
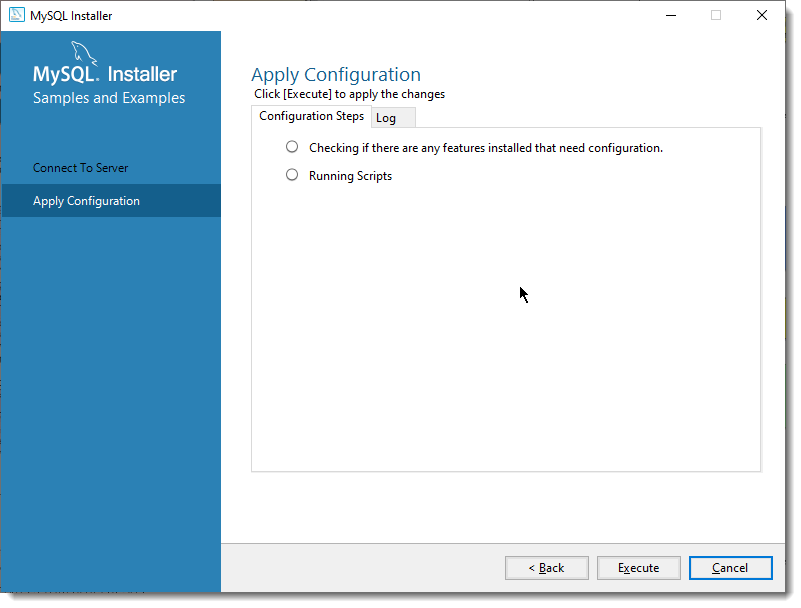
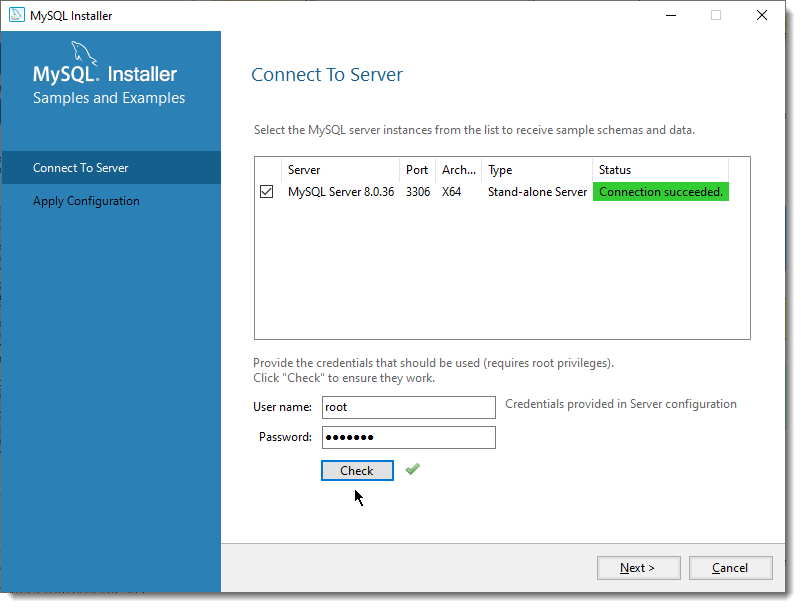
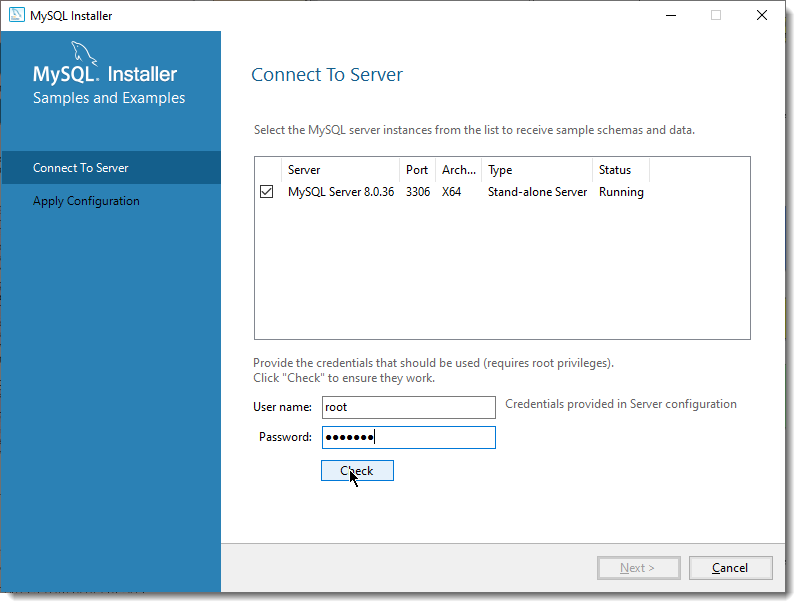
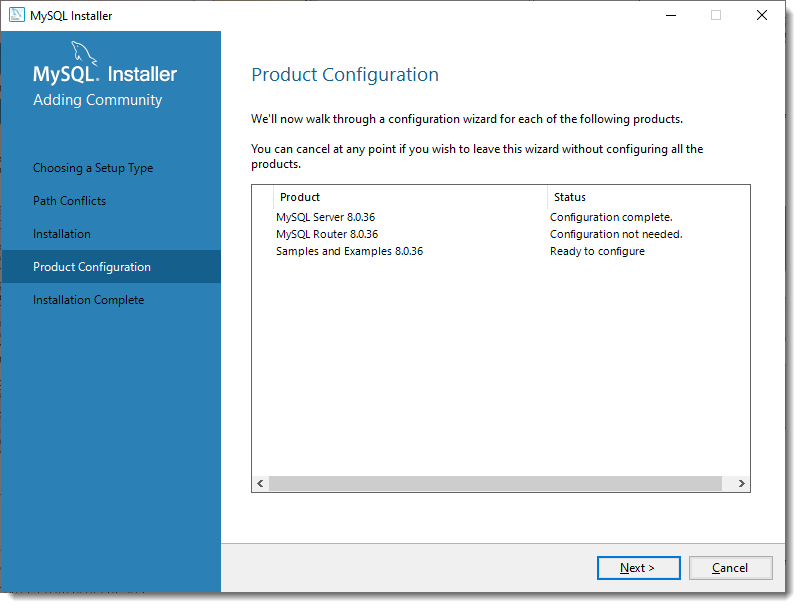
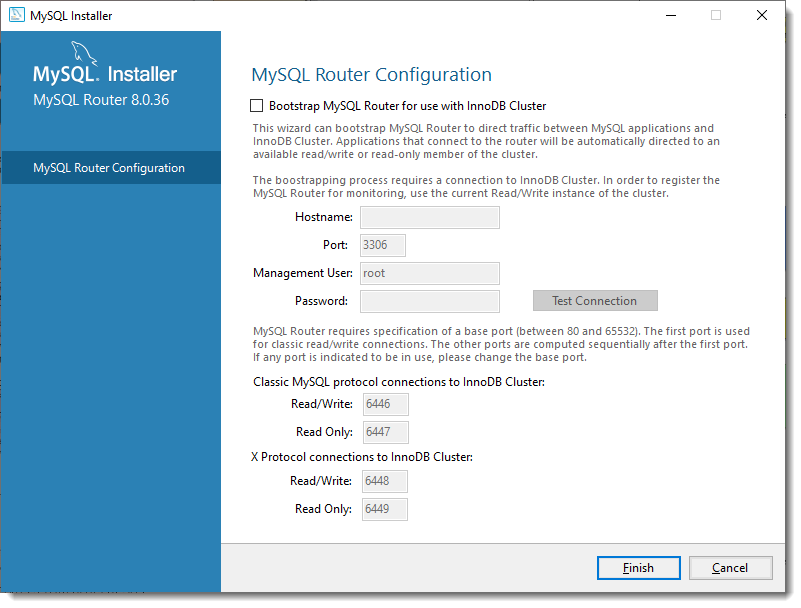
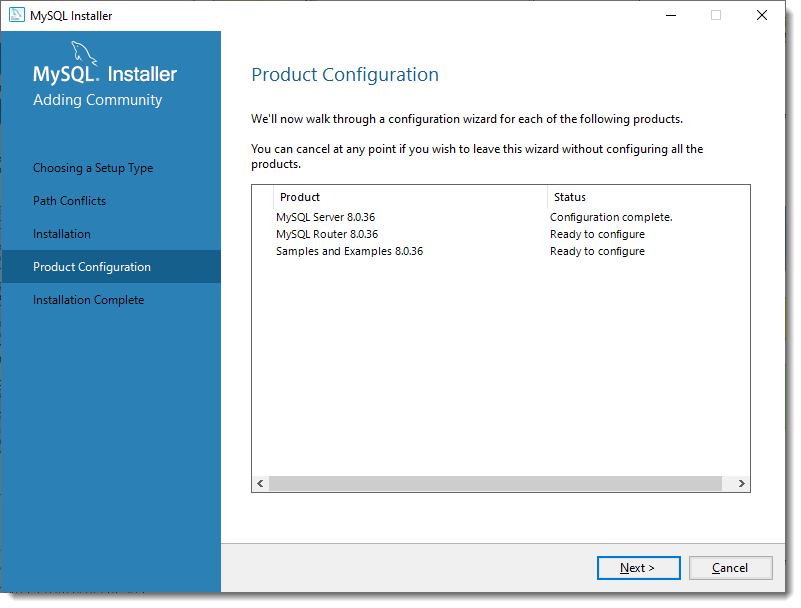
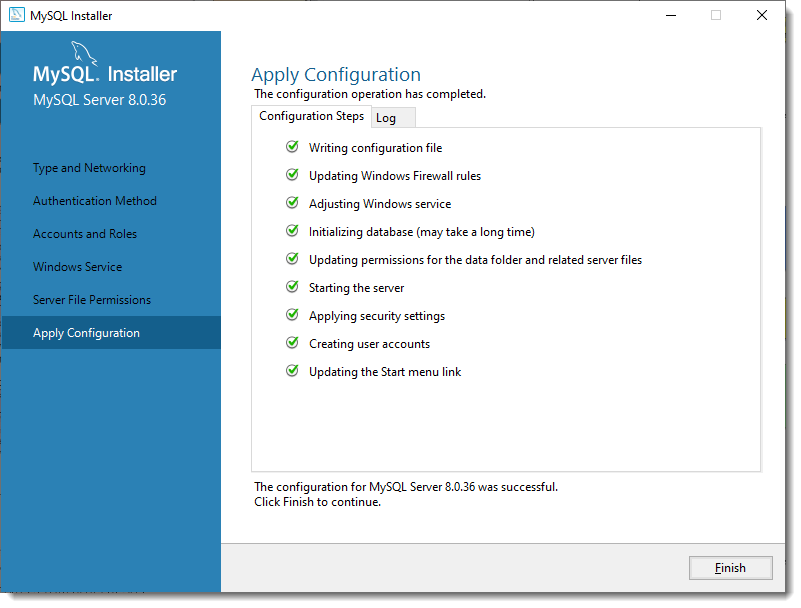
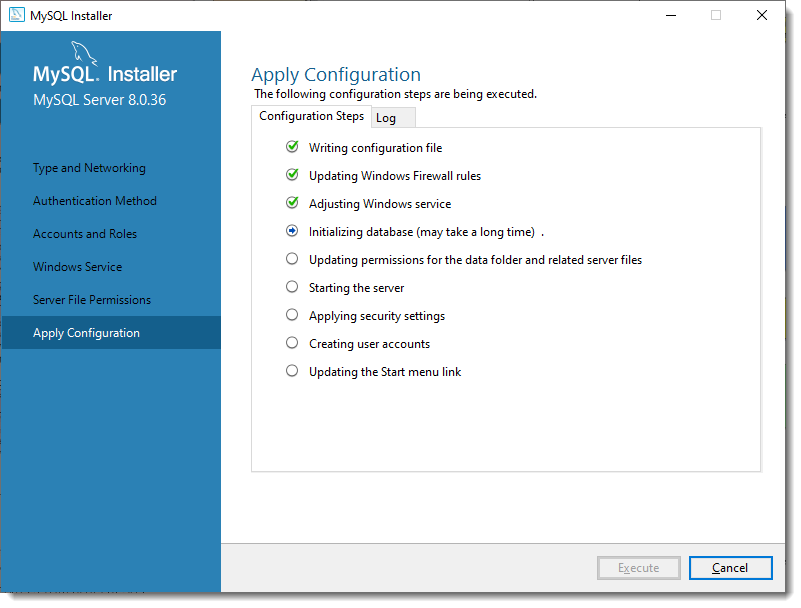
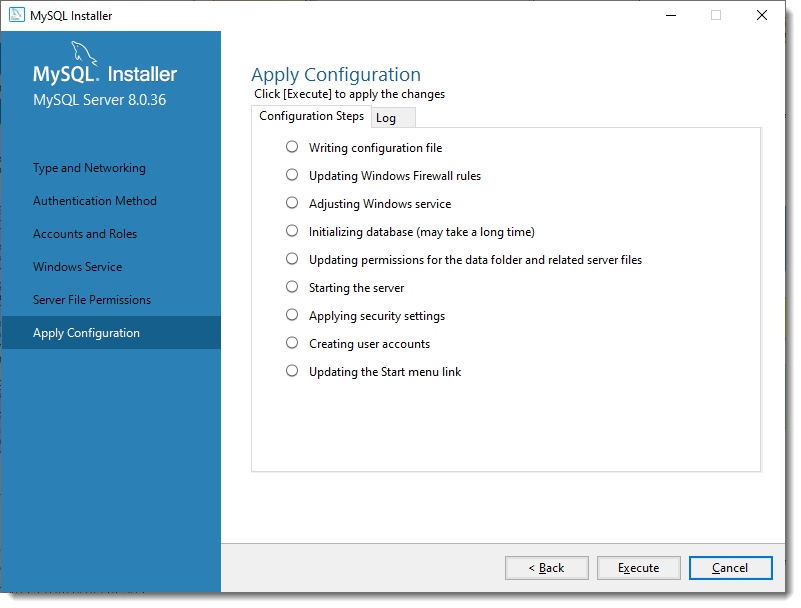
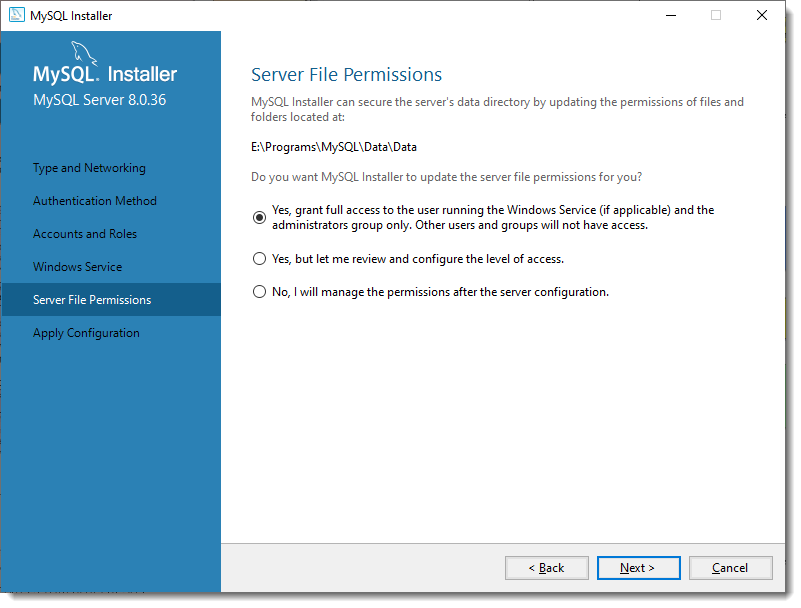
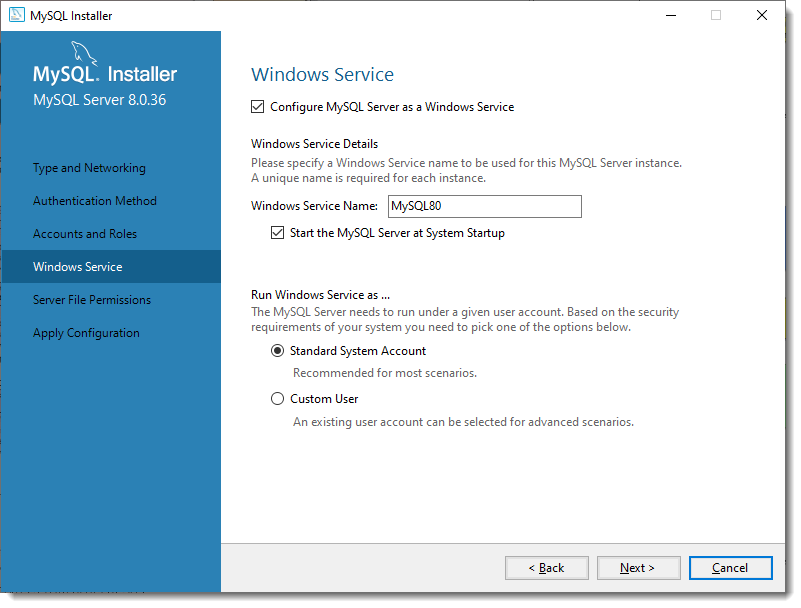
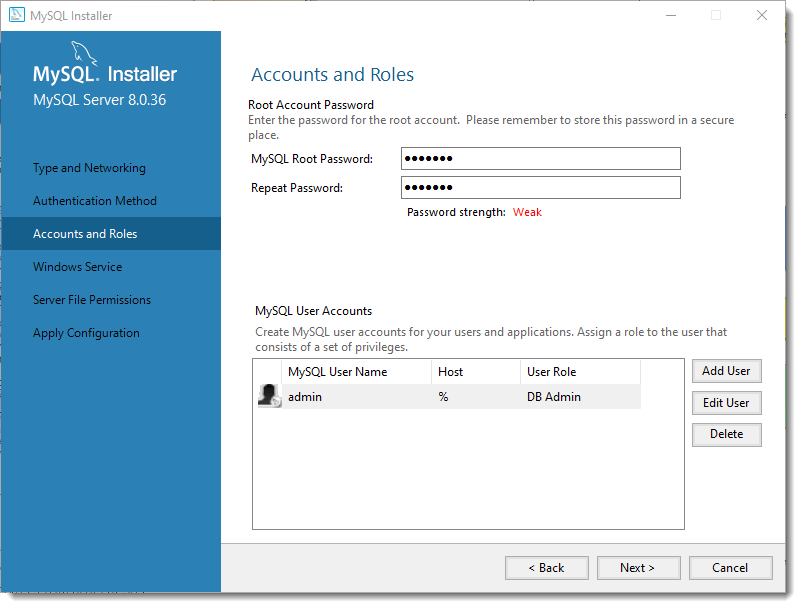
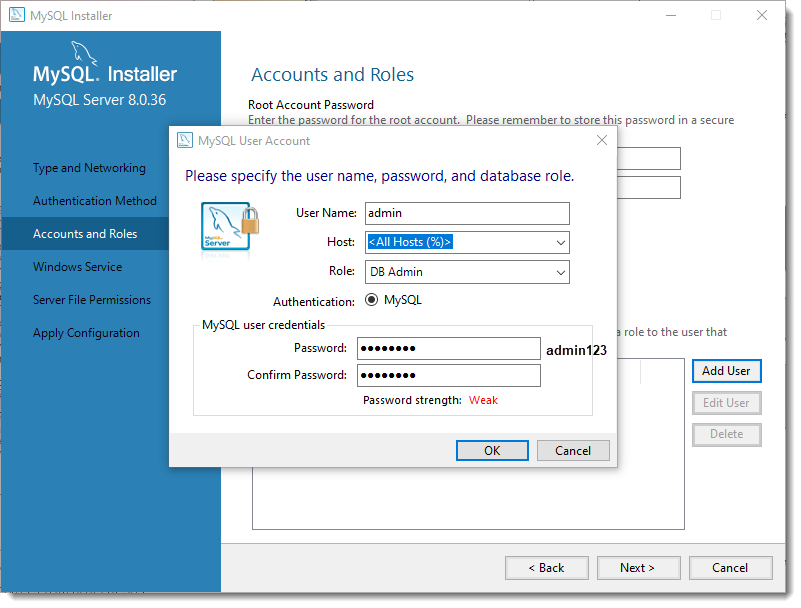
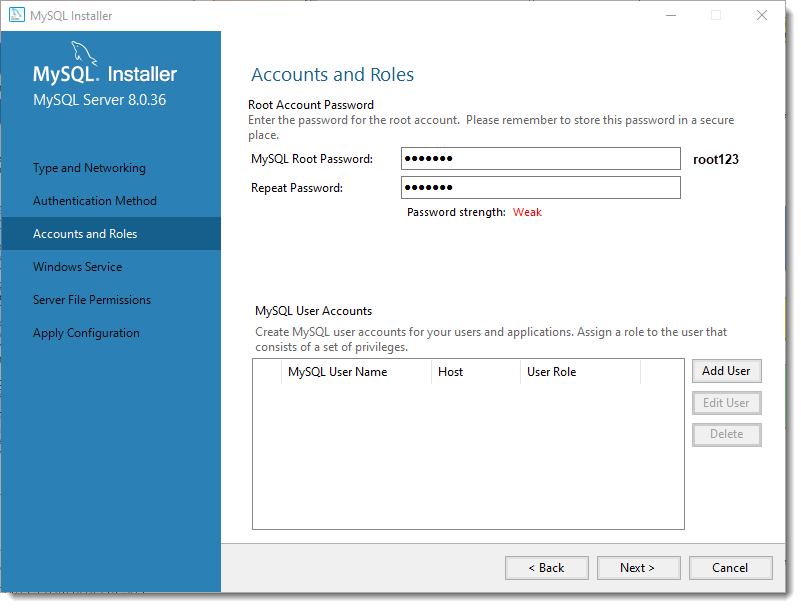
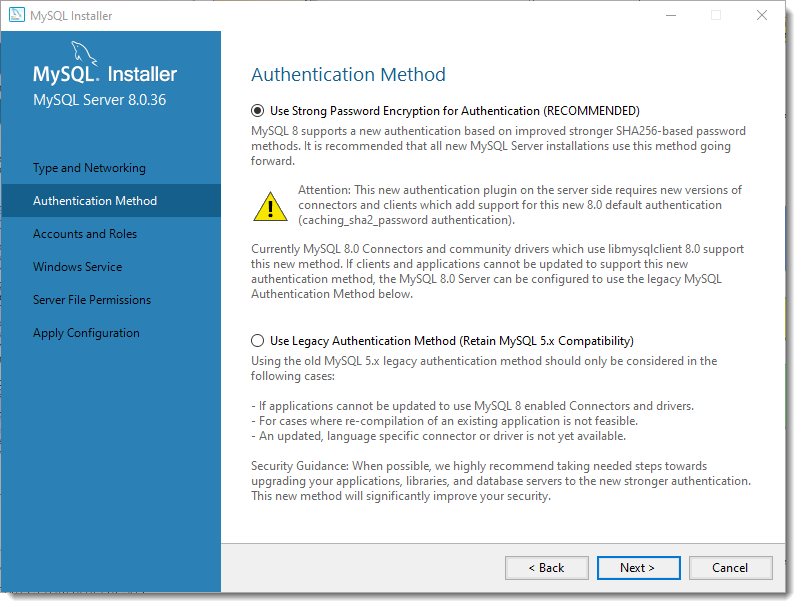
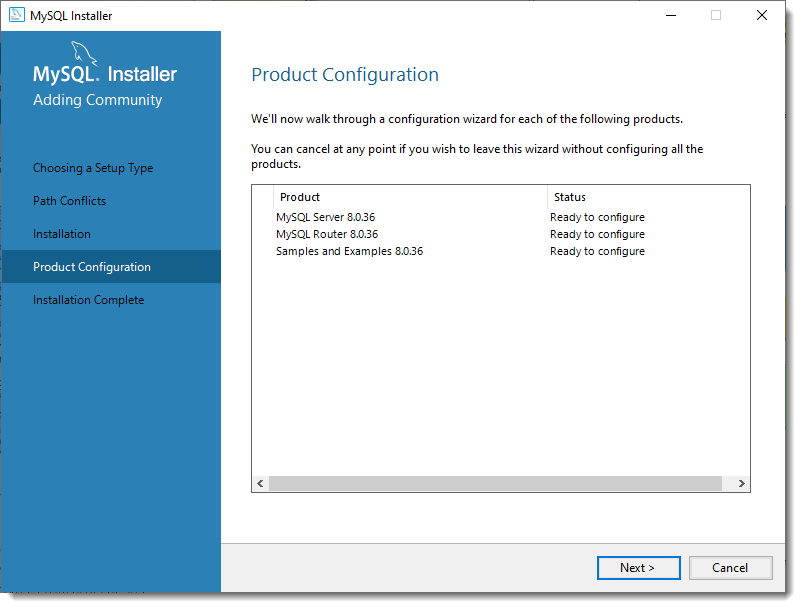
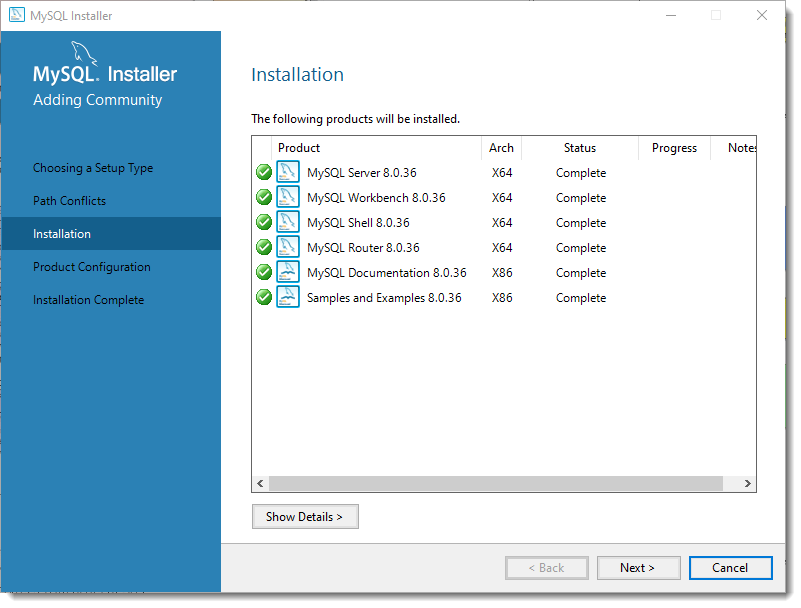








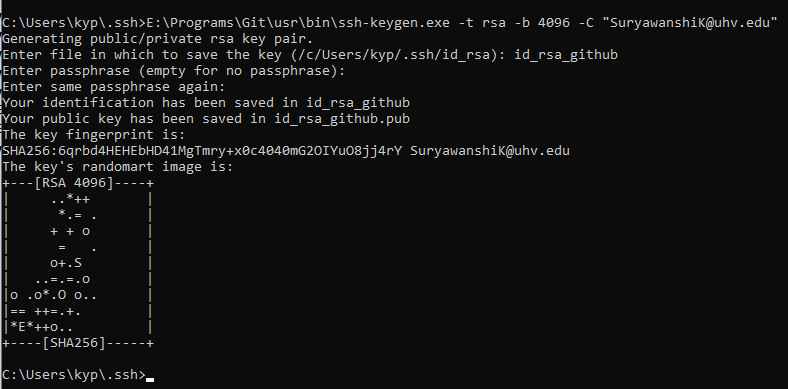


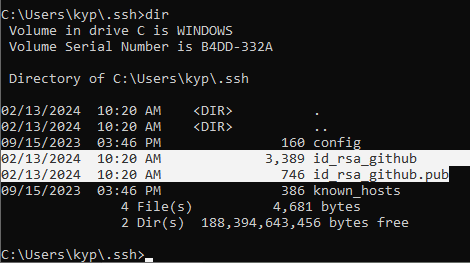


## Setup Git <->GitHub communication

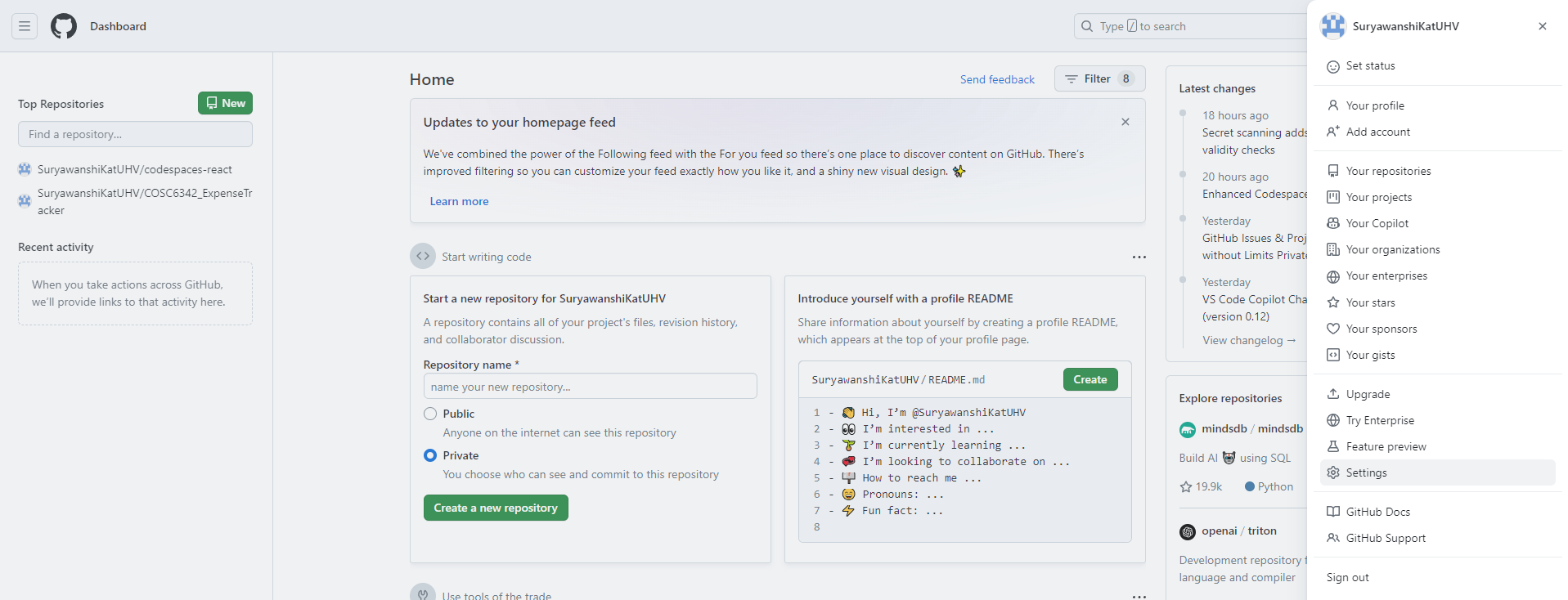
To push/pull code from GitHub to local Git repository we need to configure secure connection using private-public key pair.

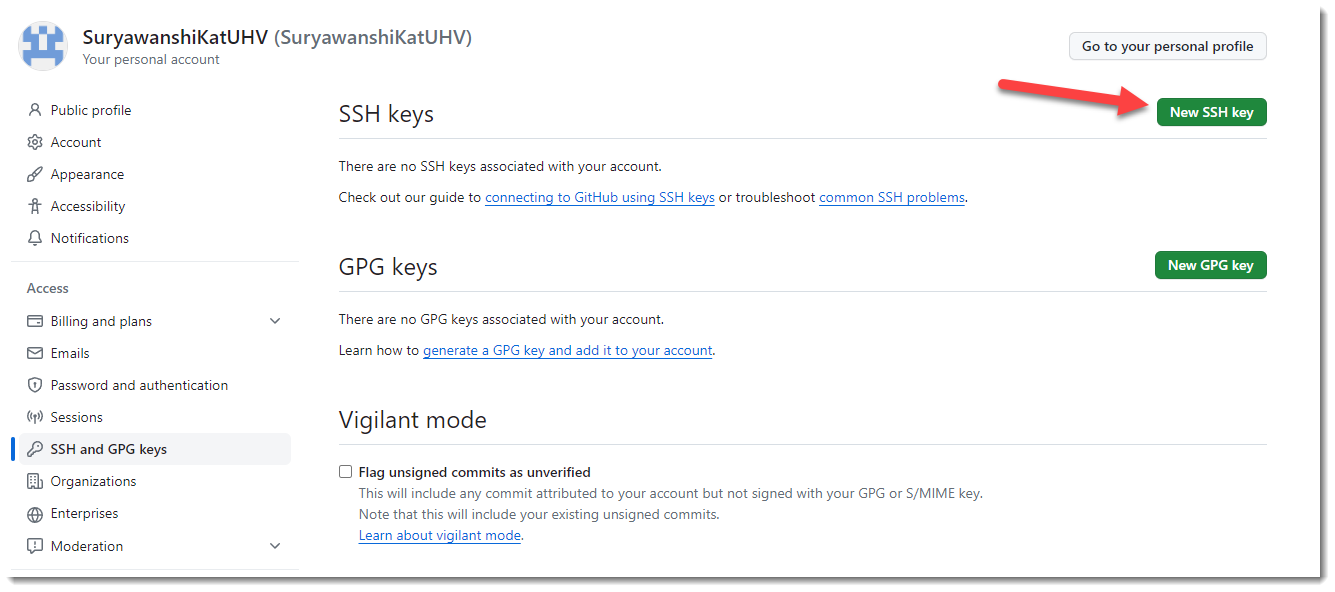
Navigate to the user profile directory and create a ssh key. For this I have used ssh-keygen application from Git installation.



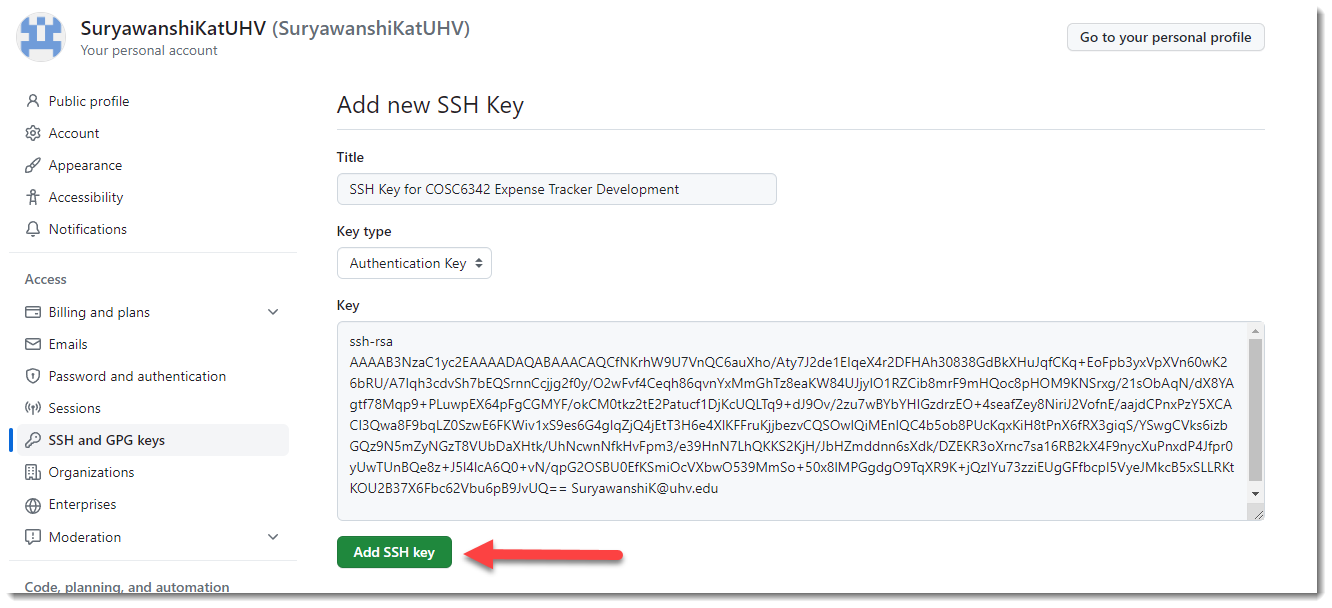


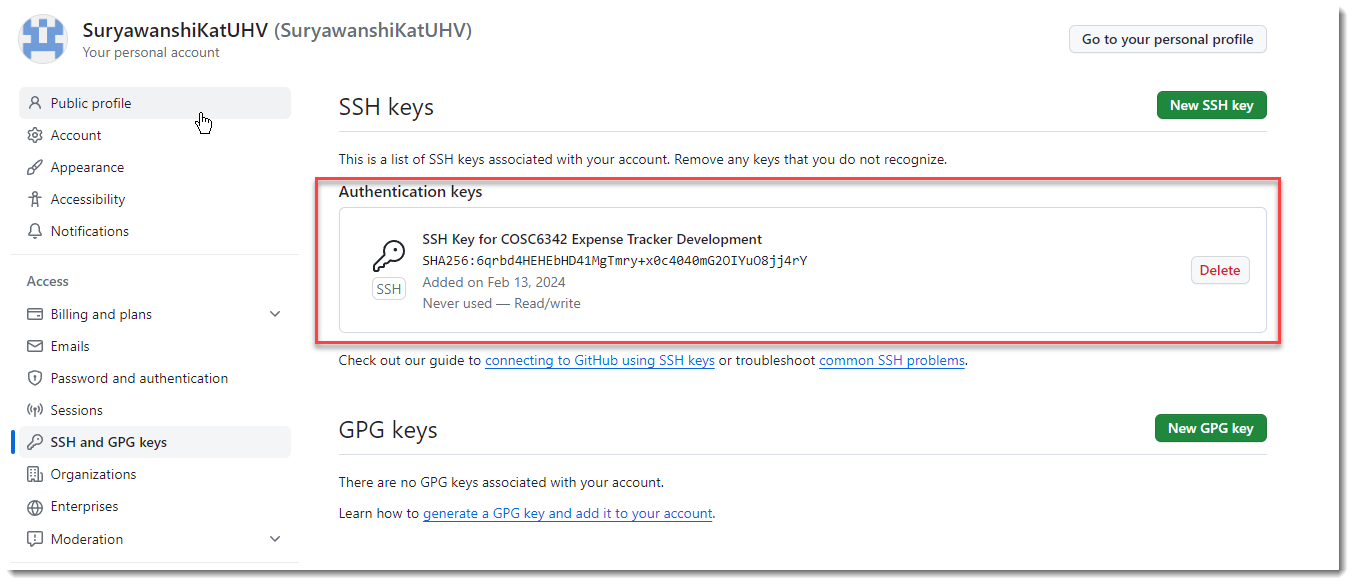
Now add the public key in Github settings.





Now copy the contents from .pub key file into new key created below. Give any title you like.



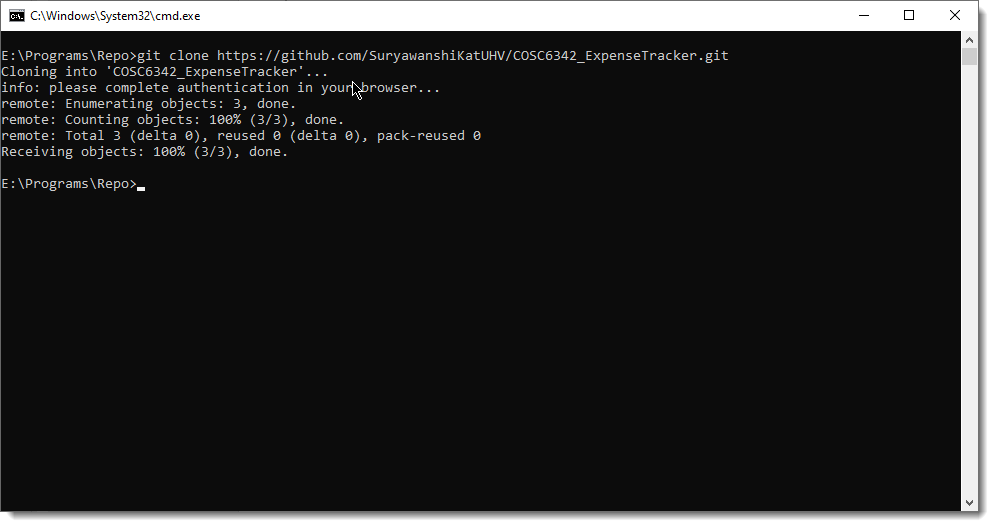
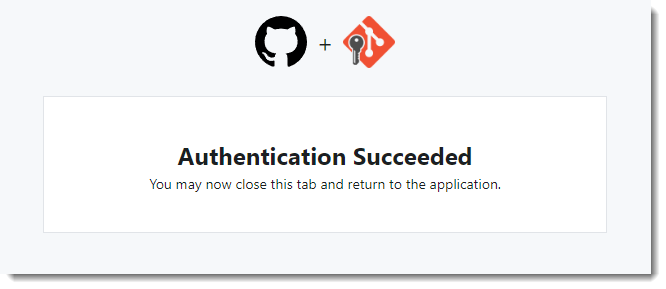
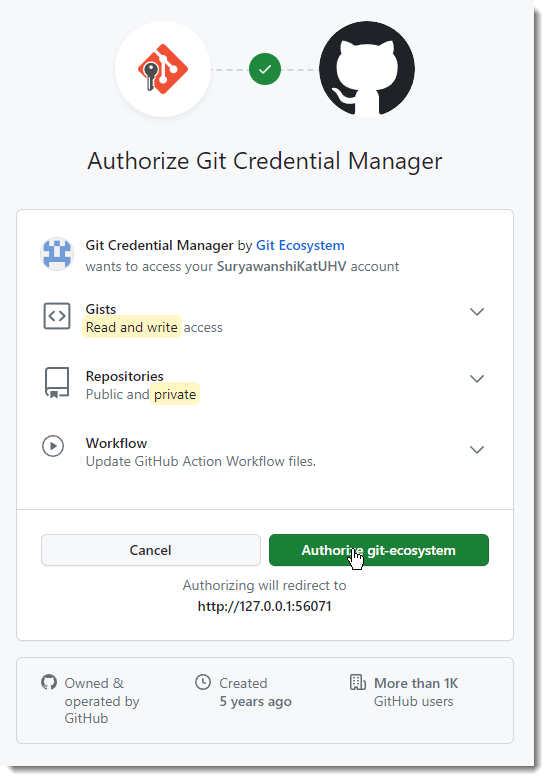


Once this is done now let’s clone and pull the repository from github into local directory to work on.

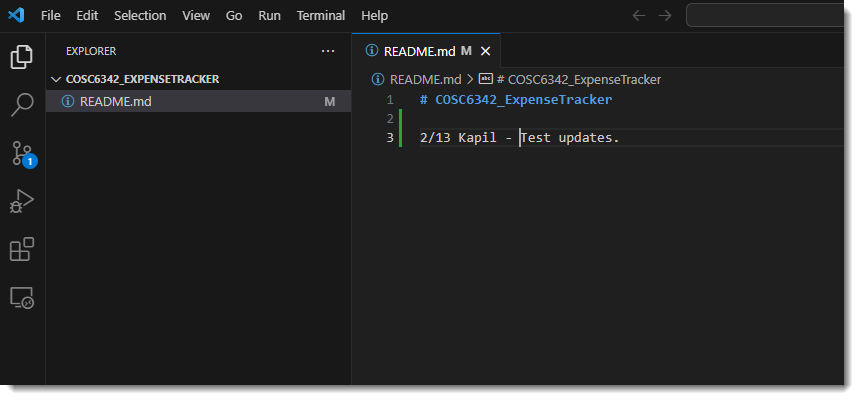
Copy the clone url for the remote repository from github.

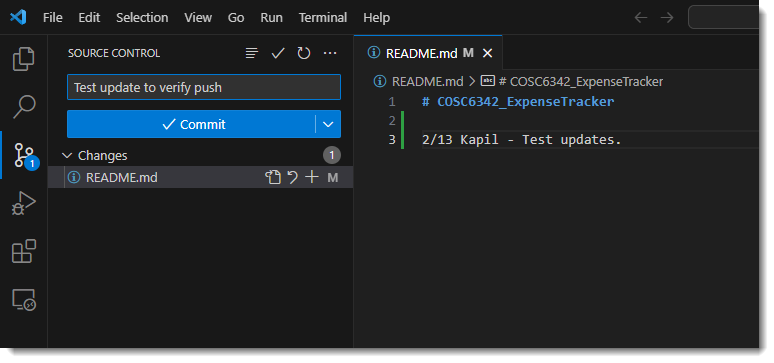


Create a local folder and open a command prompt in this folder, then clone this repo.

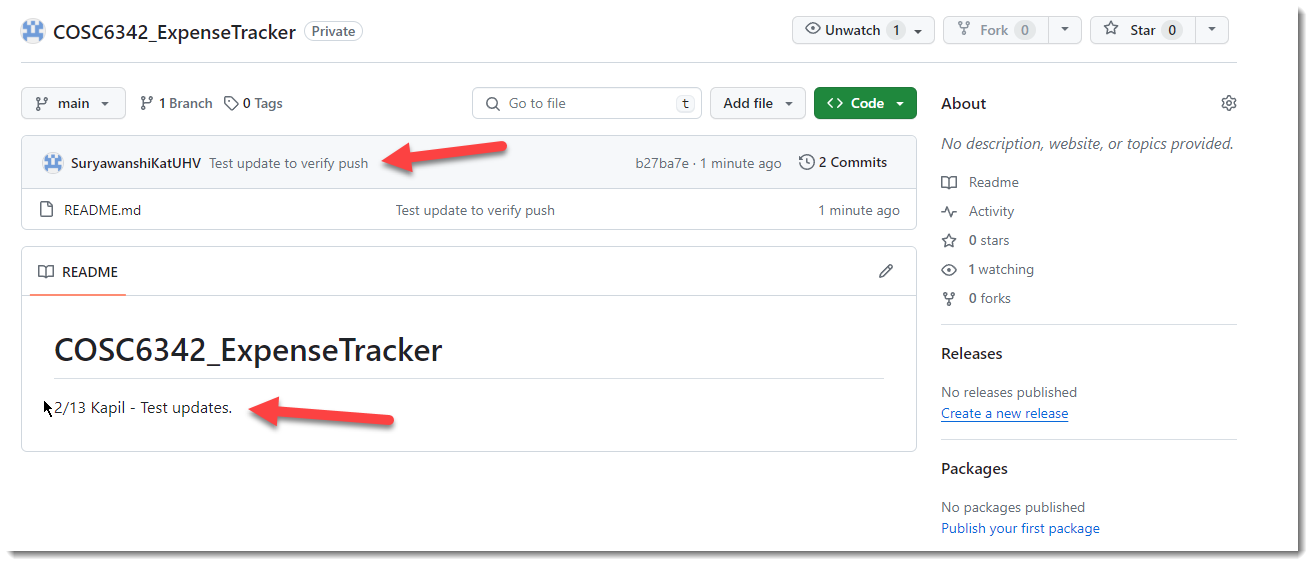


Now open the local repo folder into Visual Studio Code.

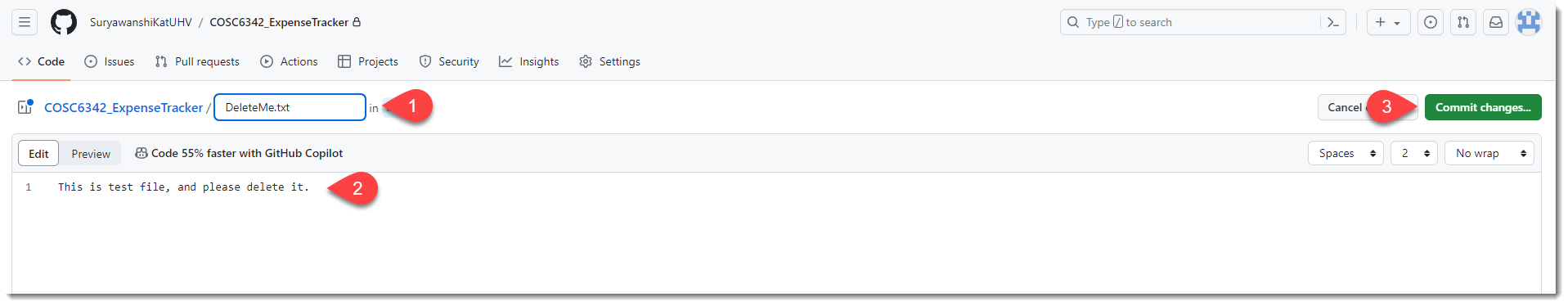
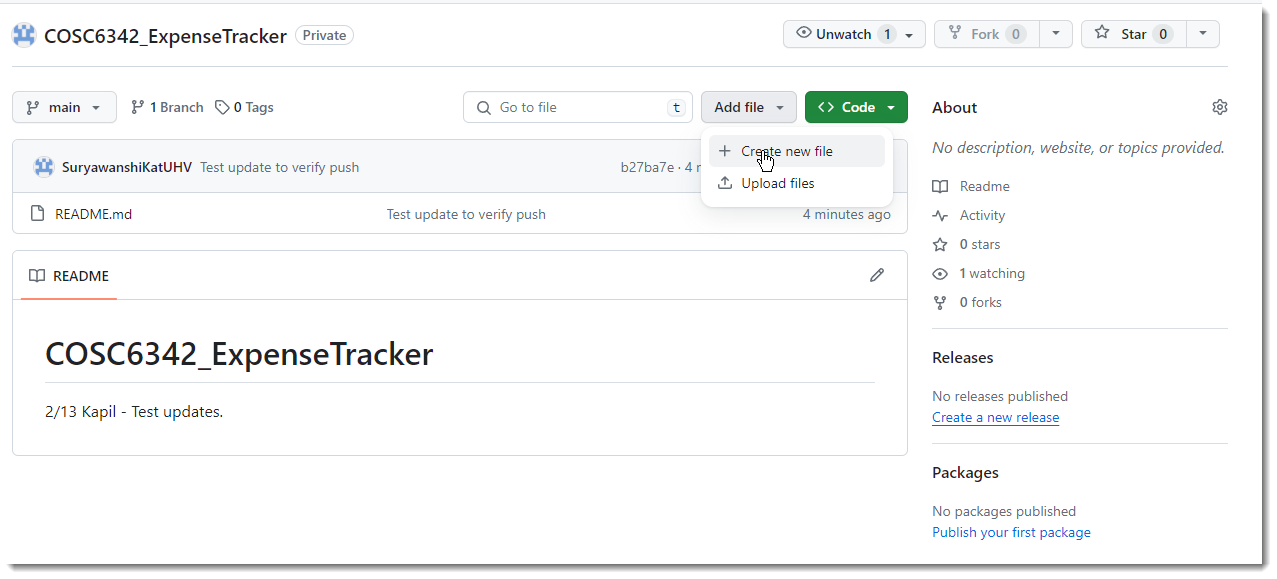




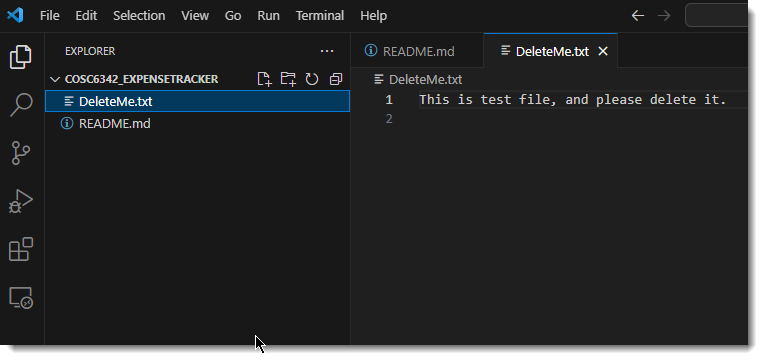
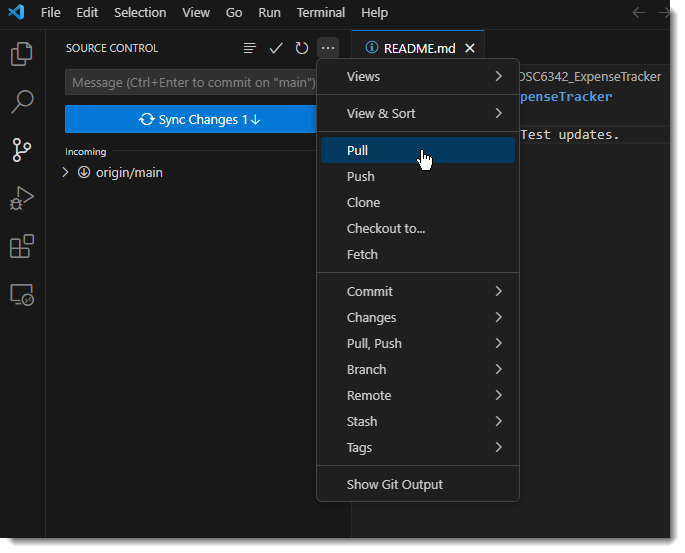
Verify that the changes are pushed into the remote repo by checking the files from Git hub.



Now creating new file in GitHub and verifying it in Visual Studio Code by pulling the remote changes.

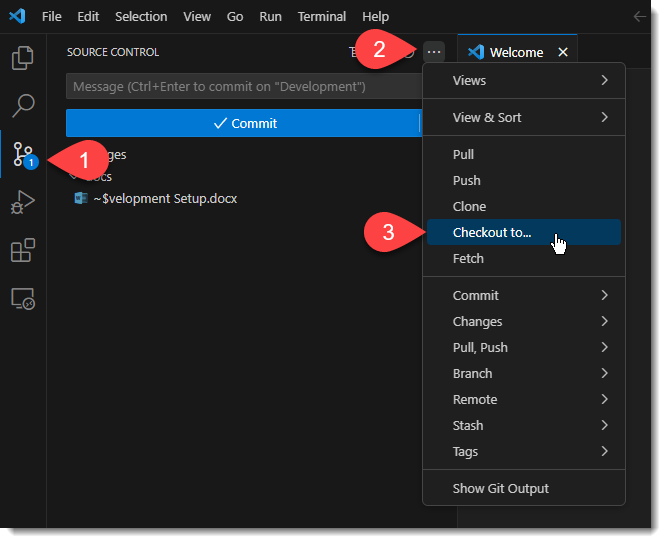


Pull changes in Visual Studio Code and confirm.

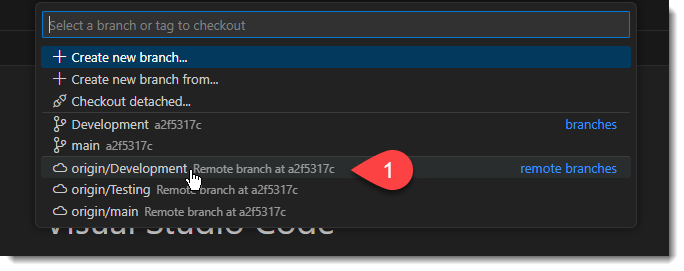


## Working with Git Development branch

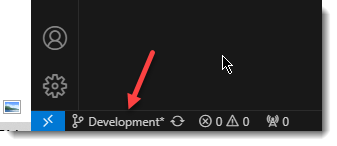
As usual open the Visual Studio Code and Sync the Git repository.



Then checkout the Development branch as shown.

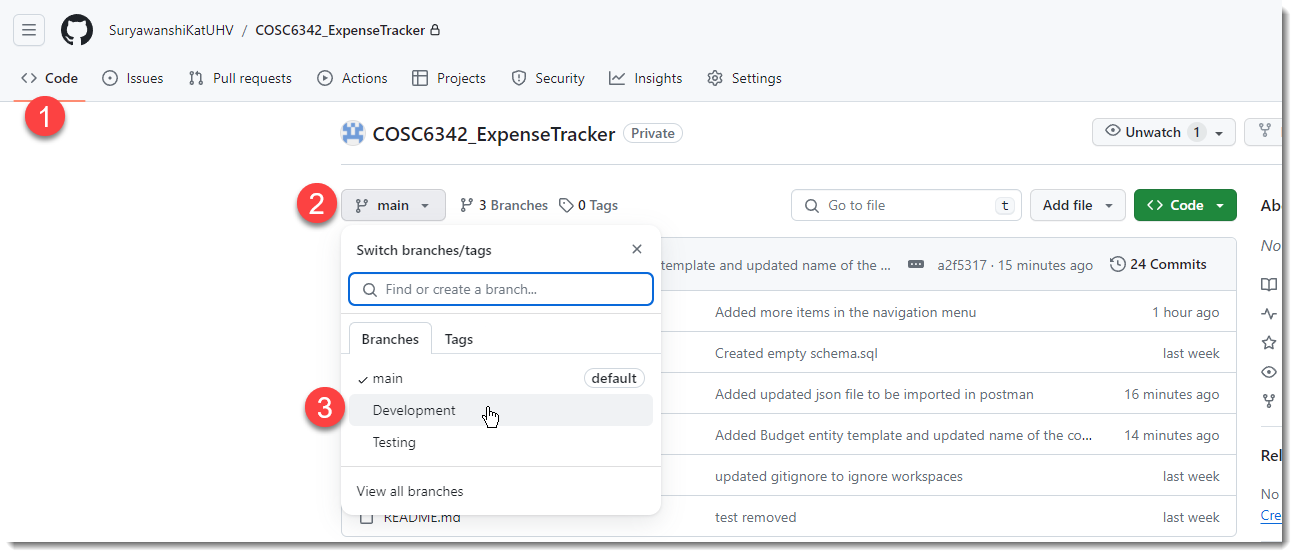


When the VSCode has synced the Development branch then the name of the branch will be shown in the lower status bar.



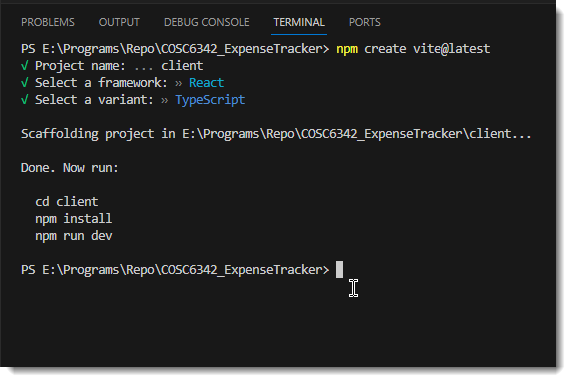
Here onwards whatever changes you commit and push will be in this branch.

To review the contents from GitHub online portal, accordingly select the Development branch.

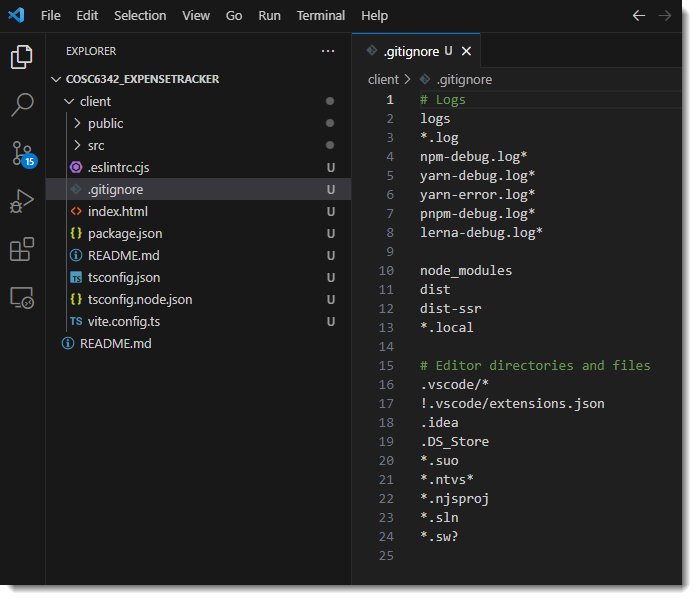


## Setup Front End app

From Visual Studio Code create a react app as below. (I have created the app and pushed it to the Github repo, so you don’t need to to these steps. Only pull the latest code into your local repo.)

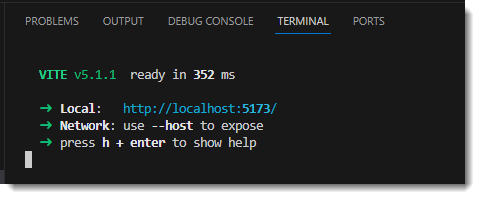


This created a template project for us. Note that it also have a file .gitignore which contains all the files which should not be part of the repo as these are runtime file.



When you would pull the code for the first time, you will need to install all the dependencies locally by executing “npm install” in VSCode terminal inside “client” folder.

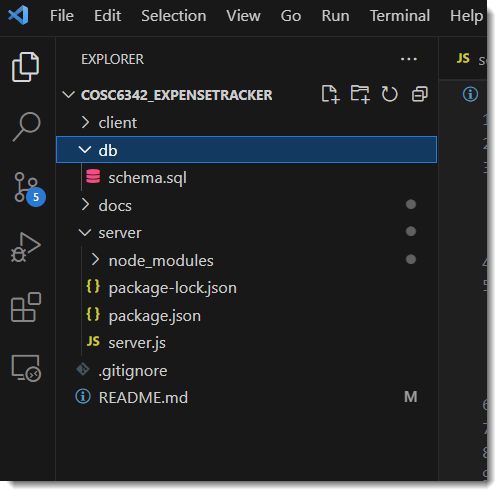
After this is done, execute “npm run dev” to start the server which can host the application in browser as below.





## Setup Back End App

Pull from Visual Studio Code which will sync server and db directories in the local repository.

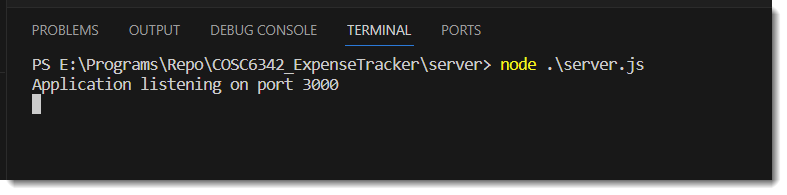


Db directory would hold the SQL script that will be used to create schema and sample data in database.

Docs directory holds documents (non-source code files) related to this application.

Server directory would hold the backend app. At this time I have server.js file created with sample endpoint services using ExpressJS module.

To run the server app, you would navigate to server directory and execute “npm install” to install related packages. Then from the terminal execute “node server.js” to start the server.



Test this server from browser by navigating to any of the following url

<http://localhost:3000/api/users/>

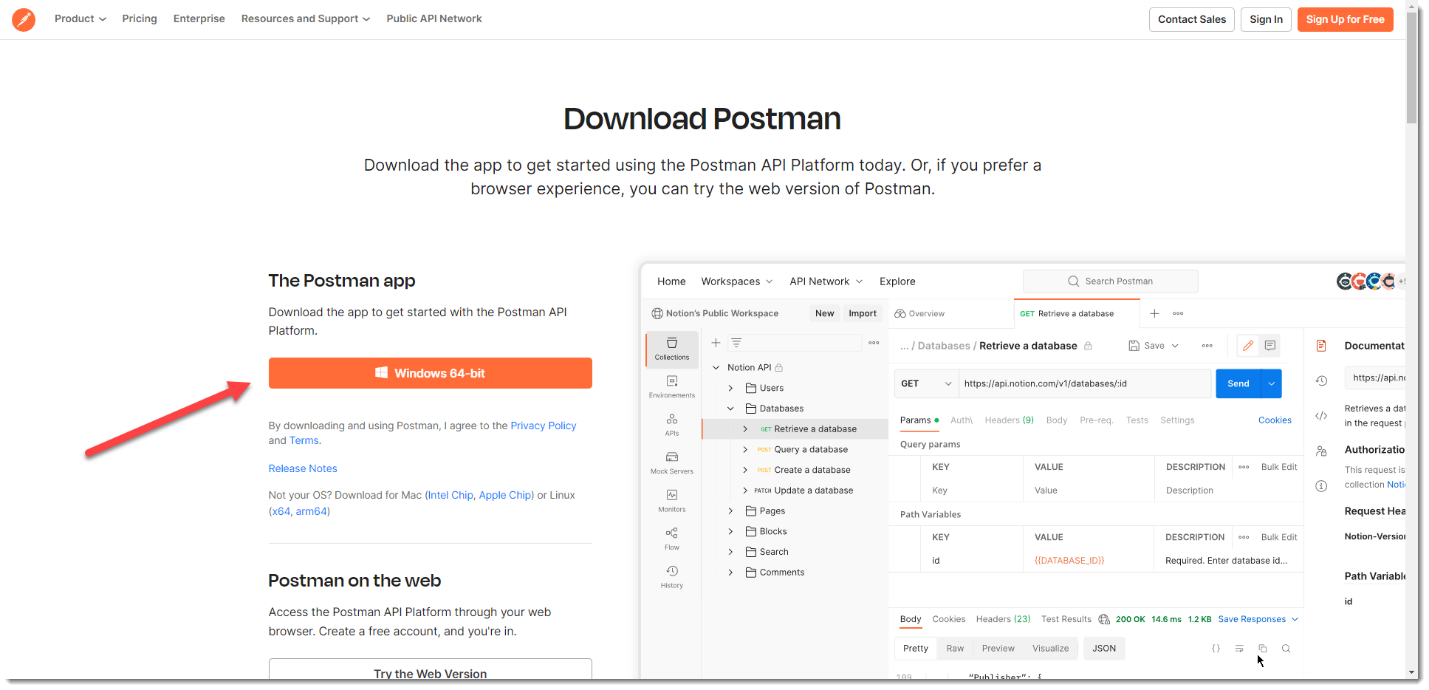
<http://localhost:3000/api/categories/>

<http://localhost:3000/api/expenses/>

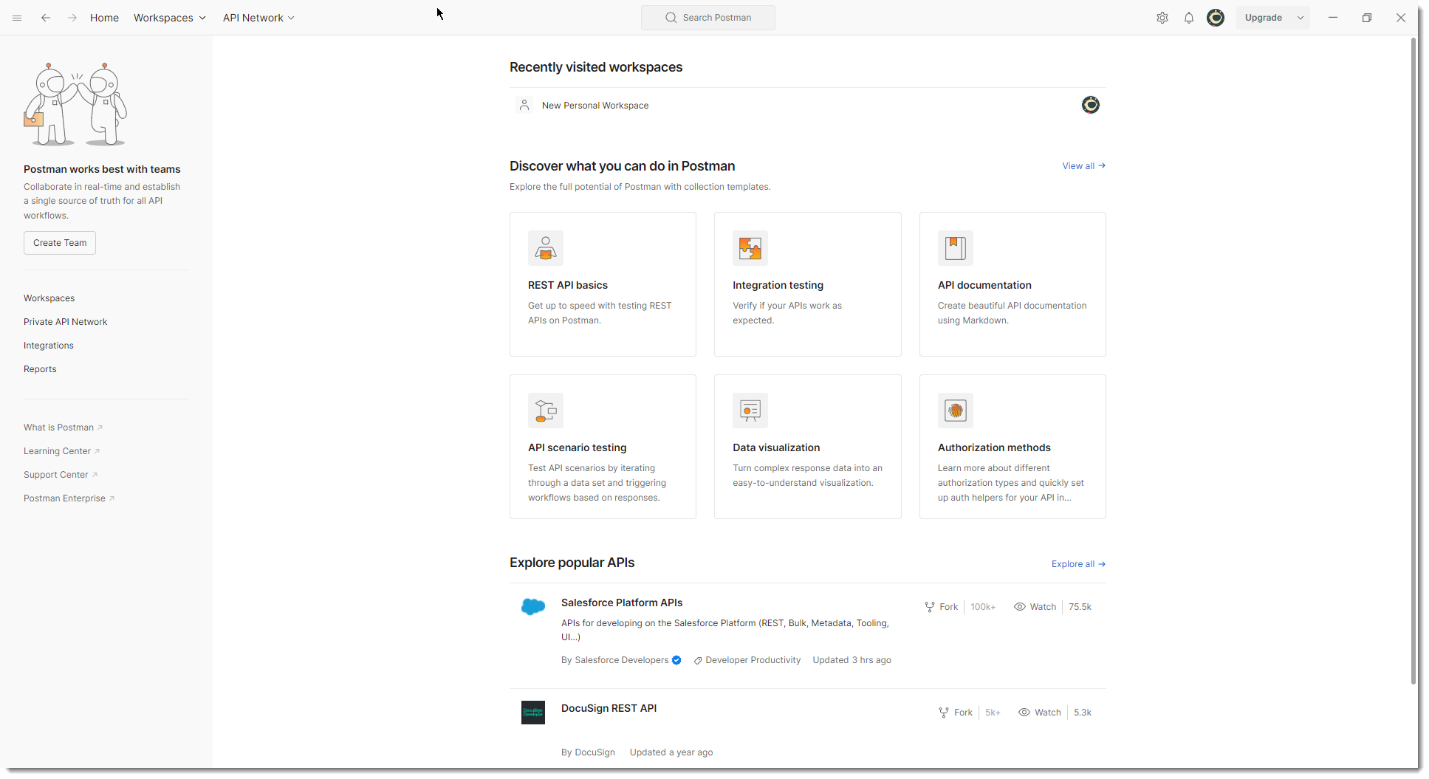
<http://localhost:3000/api/incomes/>

## Install Postman

Navigate to <https://www.postman.com/downloads/>



Download the executable and install the application.

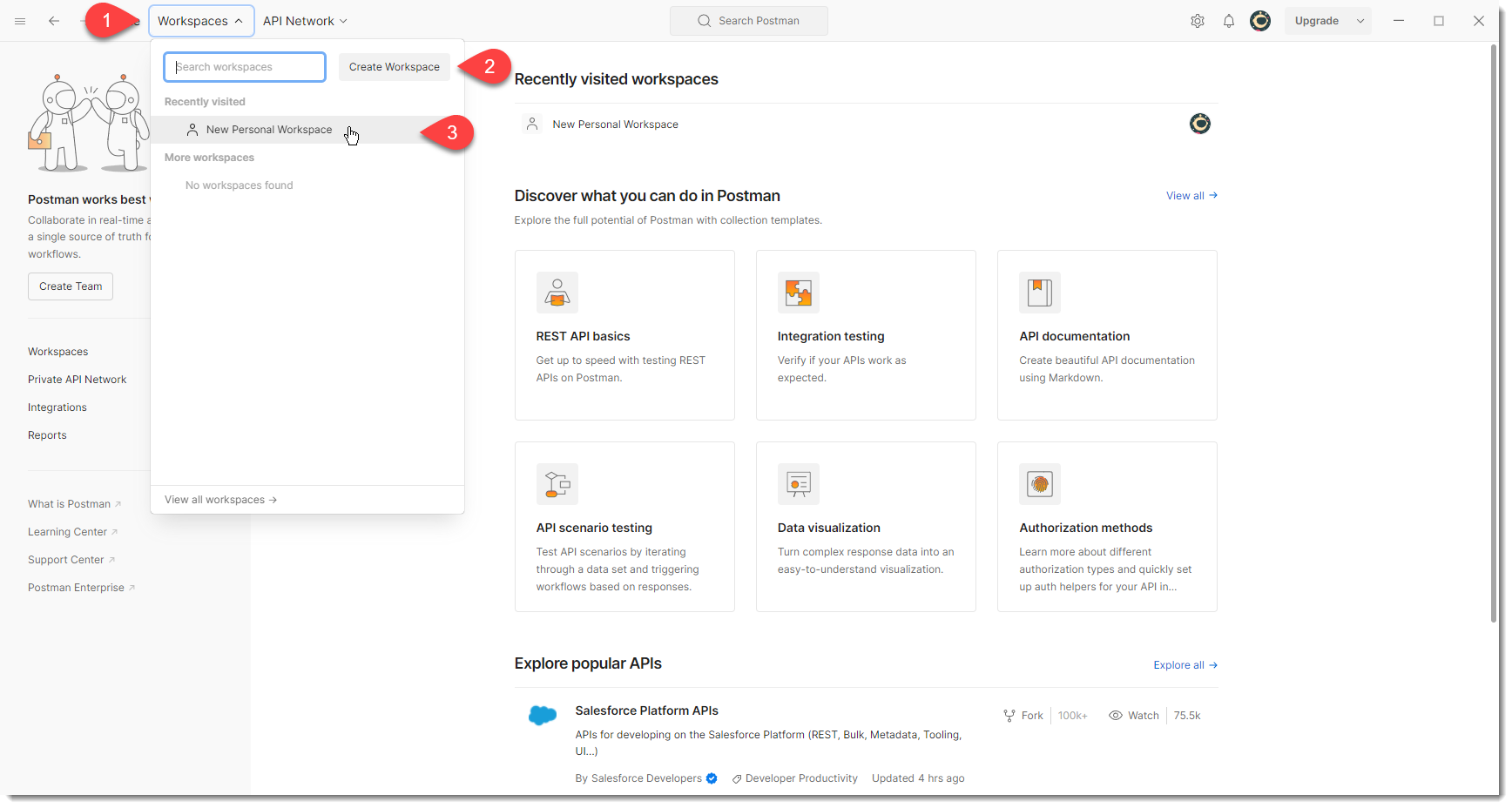


We shall use this tool to test the REST web services that we shall develop in this project.

Download the attached json file below.



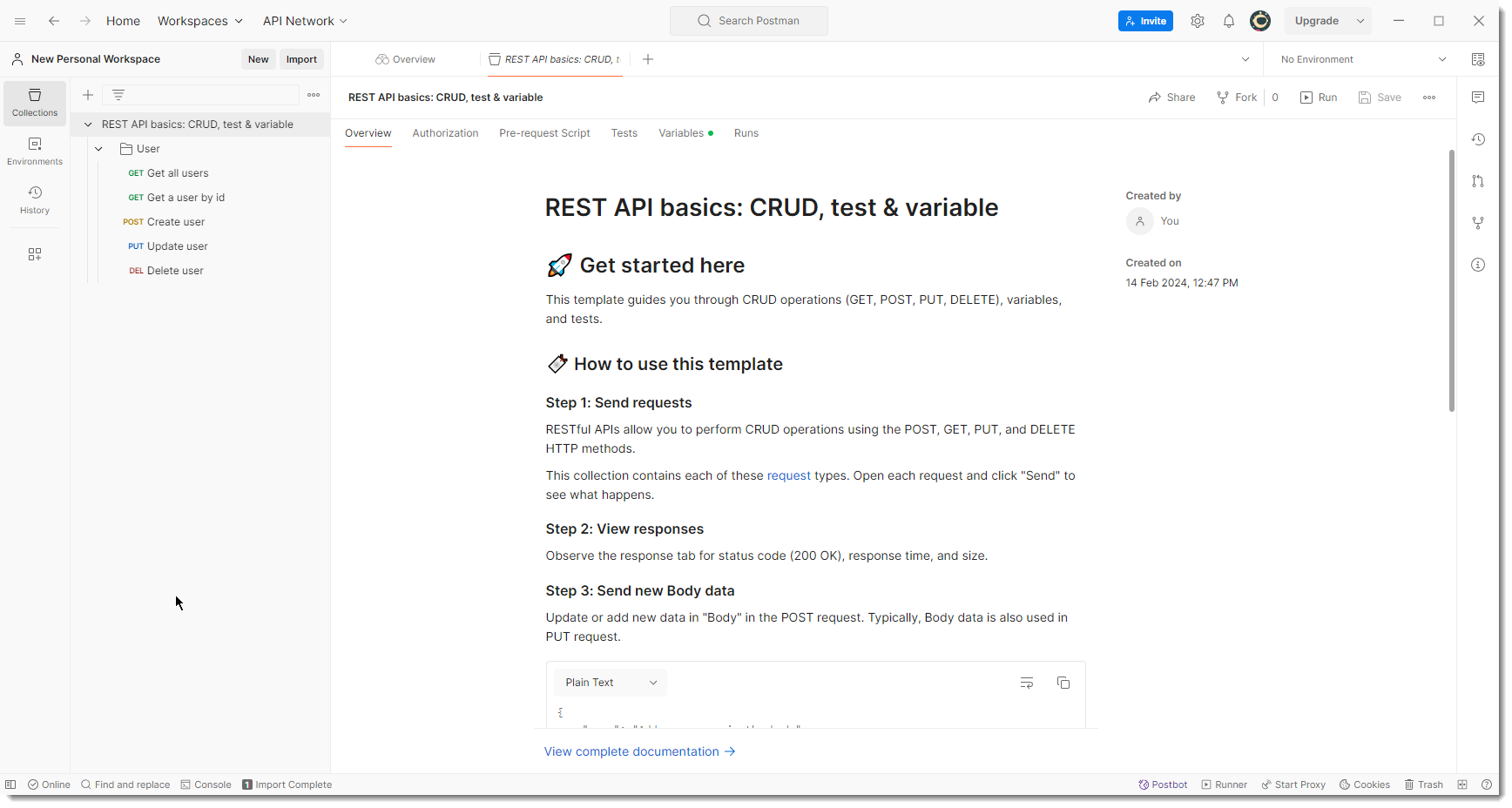
Now create and open a new workspace in Postman as shown below.



Now click import and import this json file in this workspace.

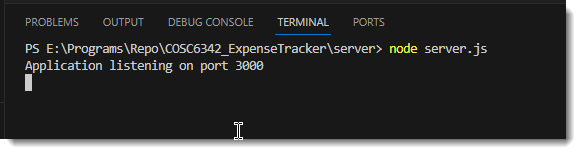


This will import a basic configuration which will be useful to test the user web service in our project.



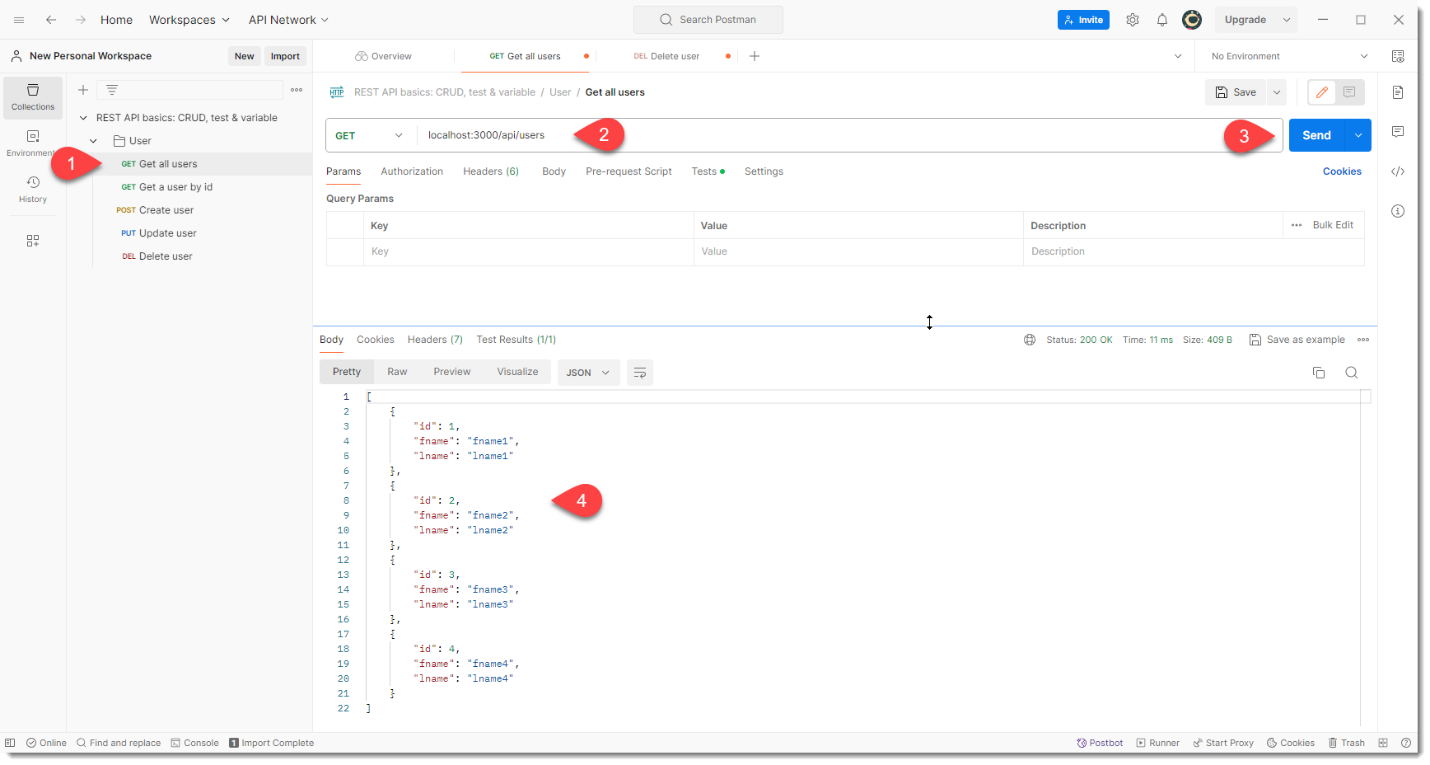
Note: I was expecting that this json file will have all the parameters and URLs I used to test the user web service, but it does not have. Therefore use the following instructions to test each of the CRUD operation.

Pull latest code from Github and then from terminal navigate inside server directory and then start the server.

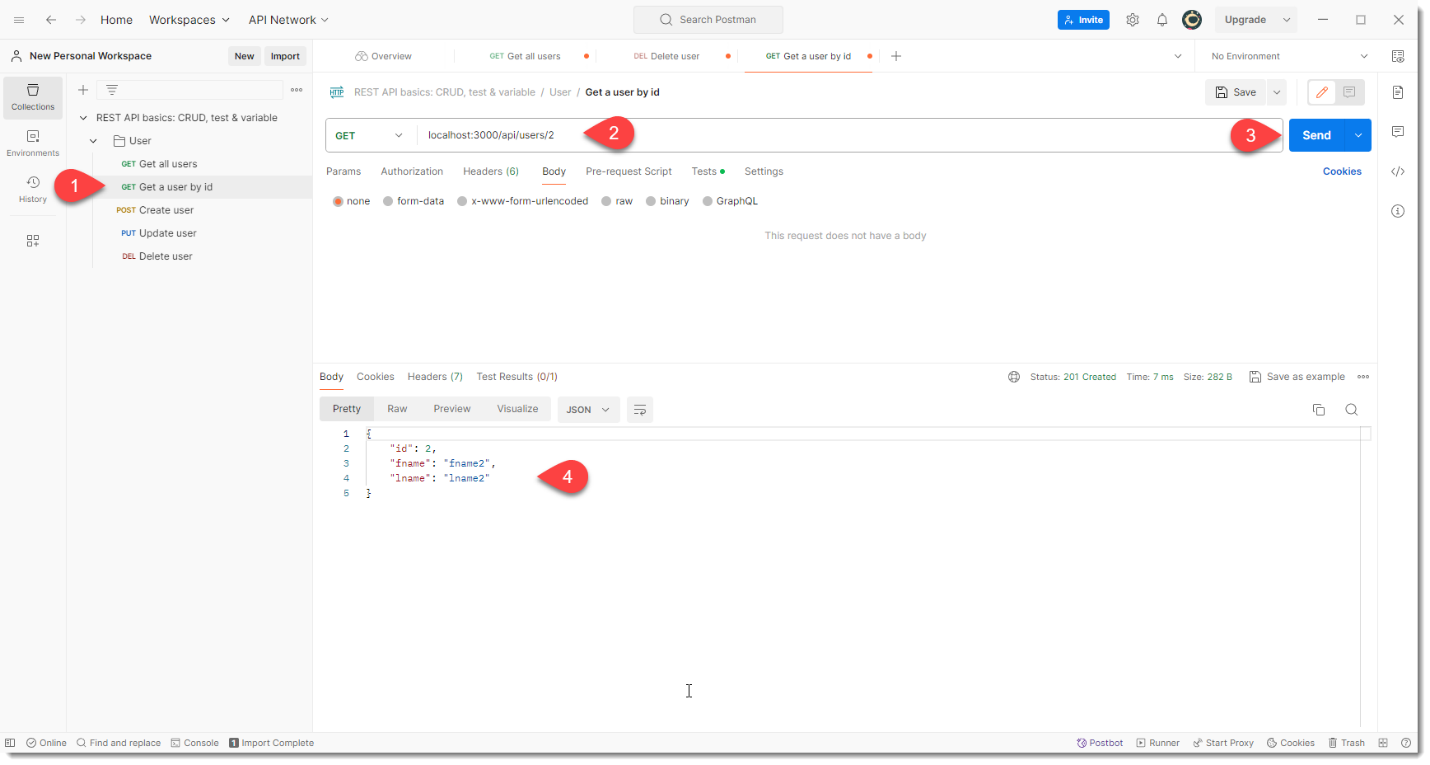


Now follow the rest of the sections to test each of the user web service end points.

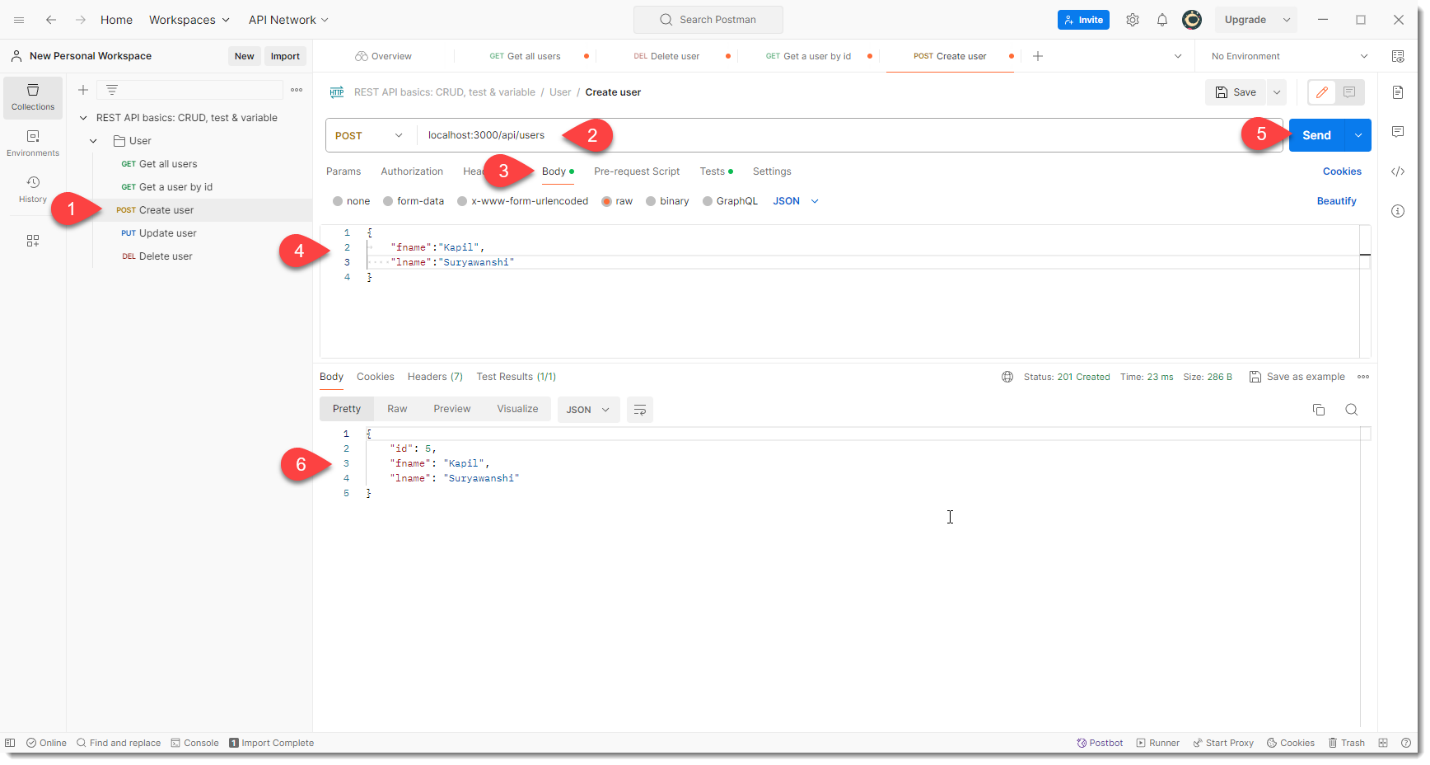
### Get all users



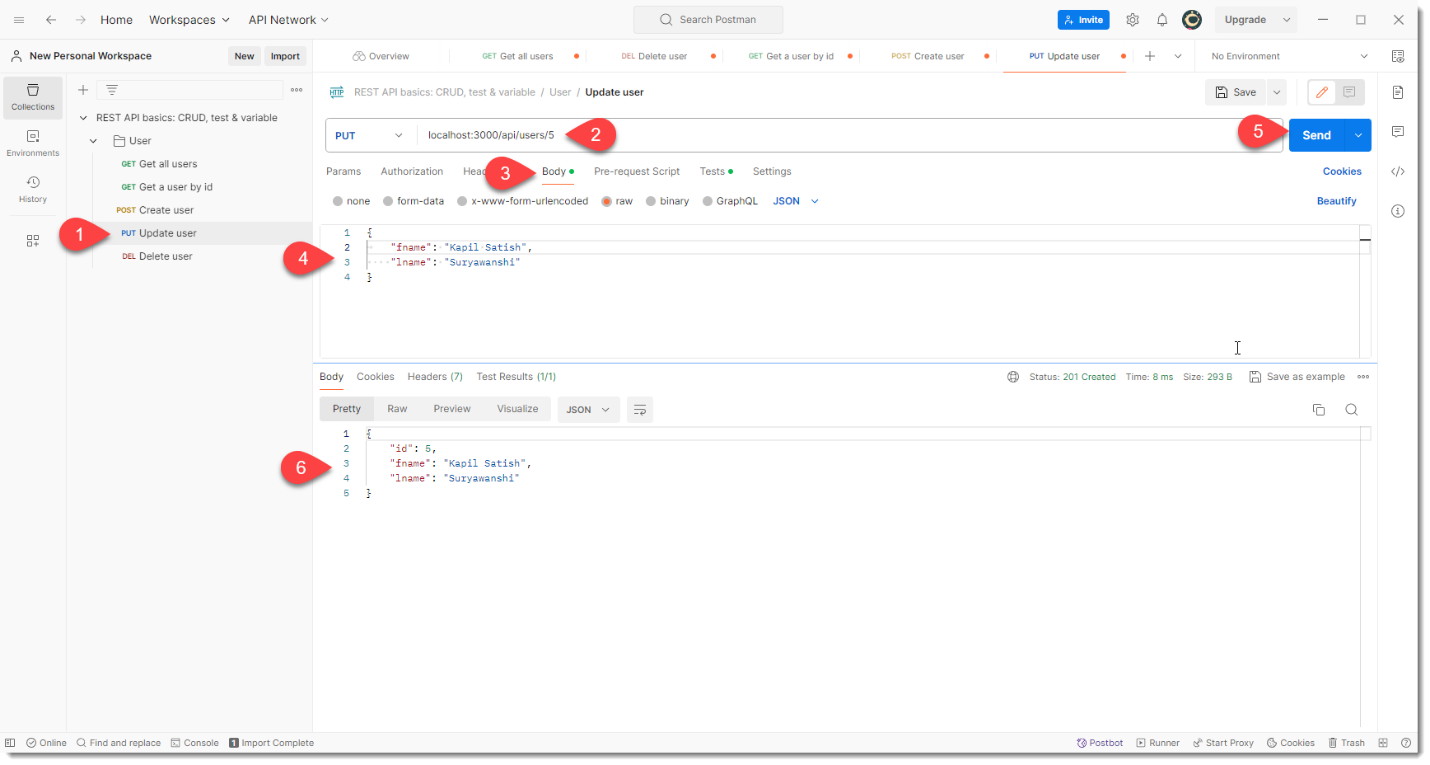
### Get a user by id



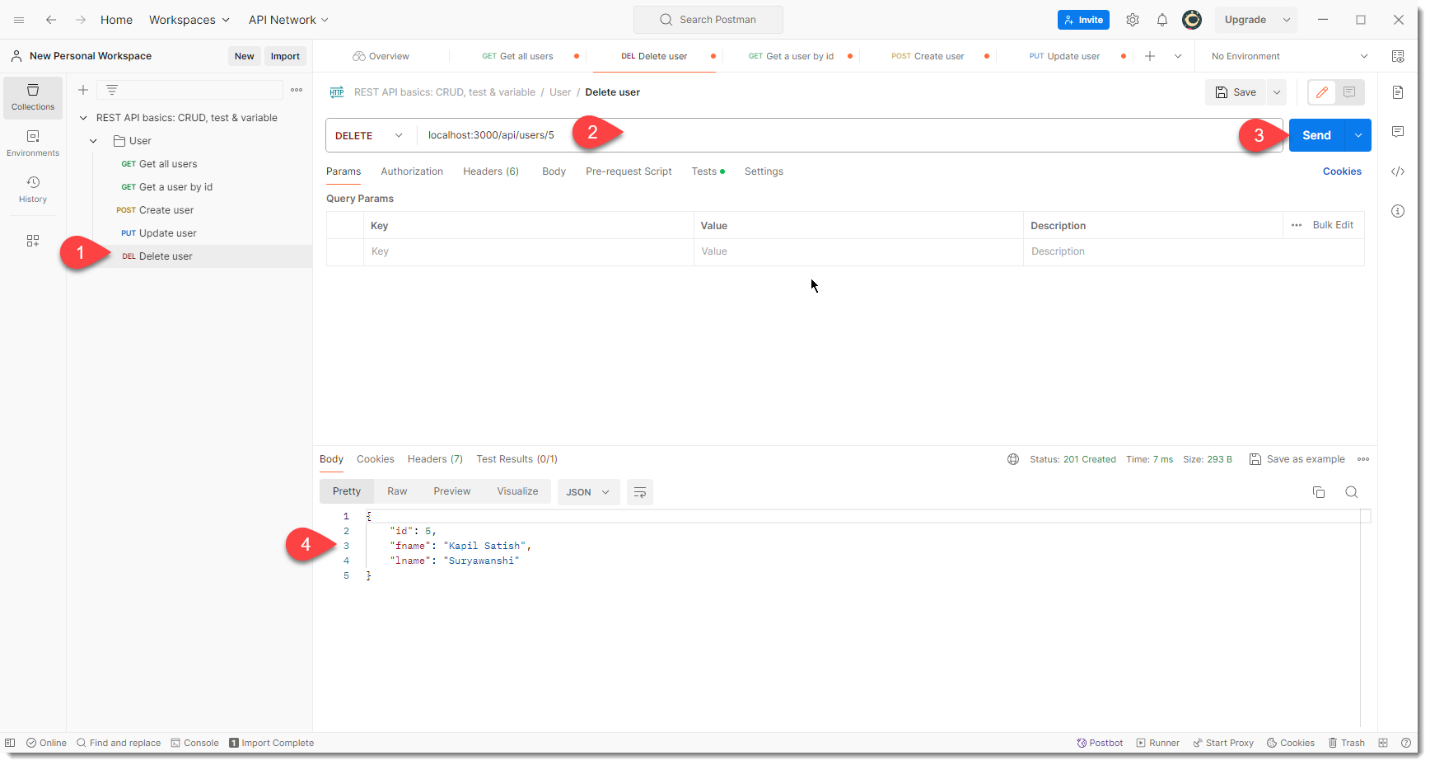
### Create user



### Update user



### Delete user



Now test the “Get all users” case again to verify that this record is deleted.

# Production Environment

# Archived