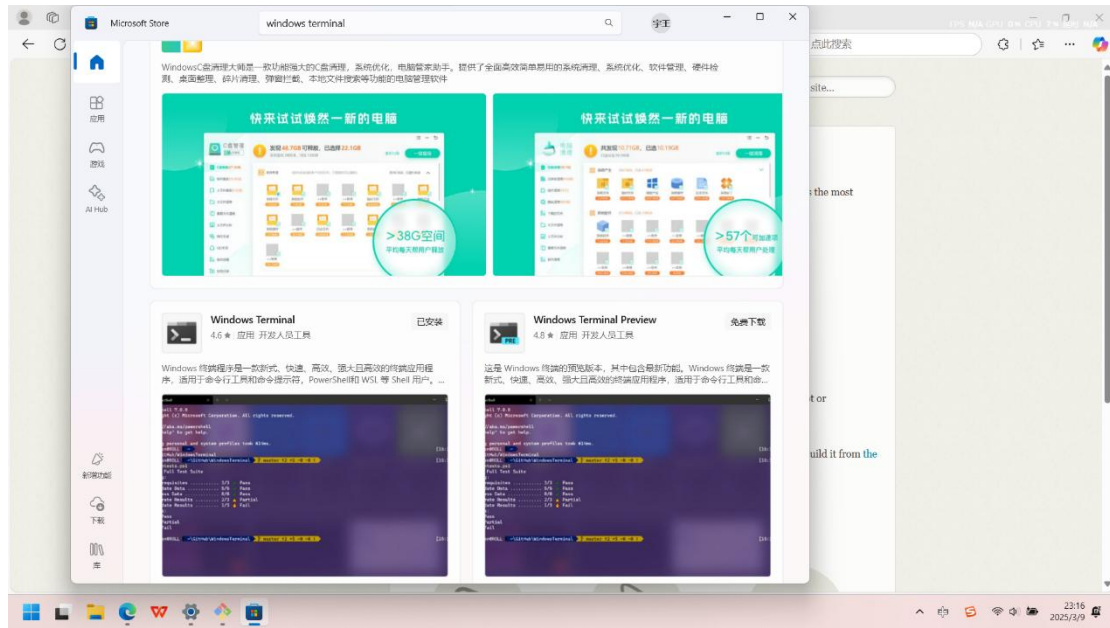
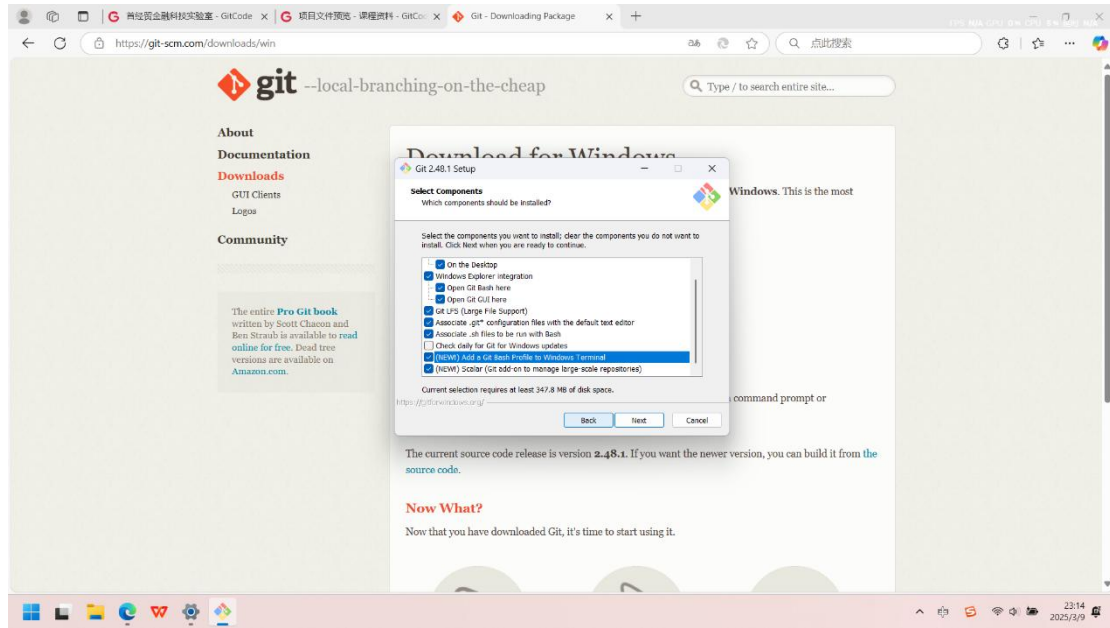
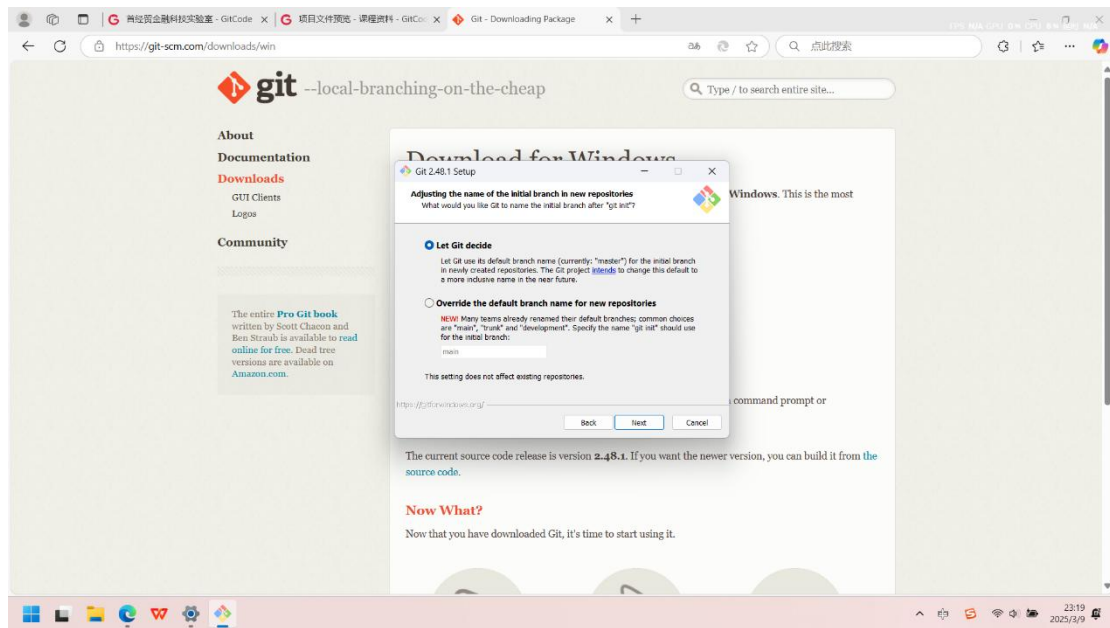
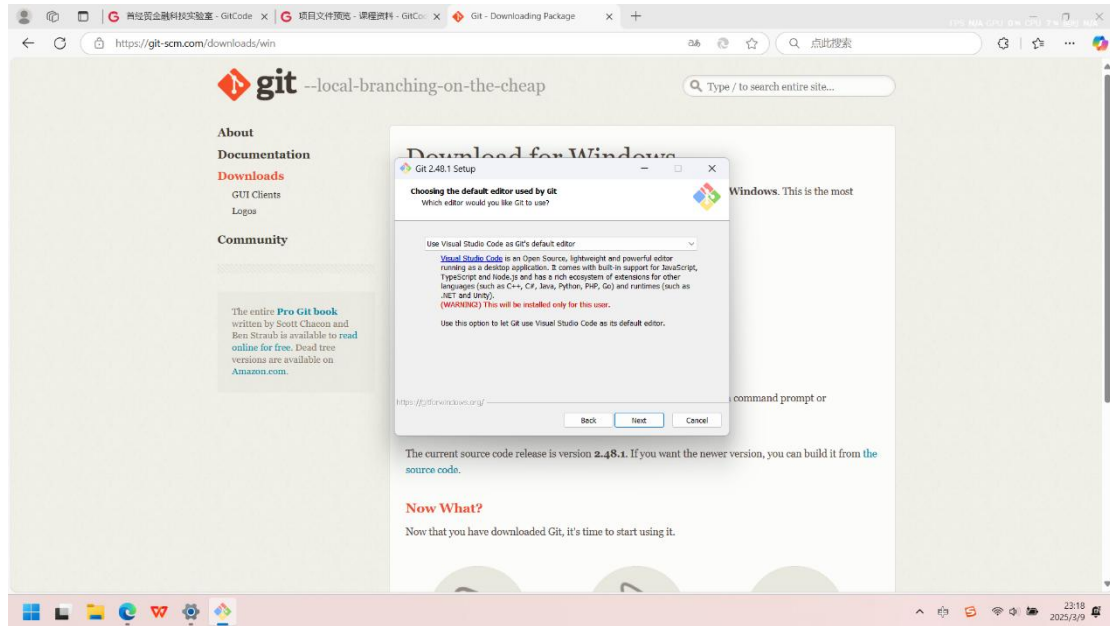
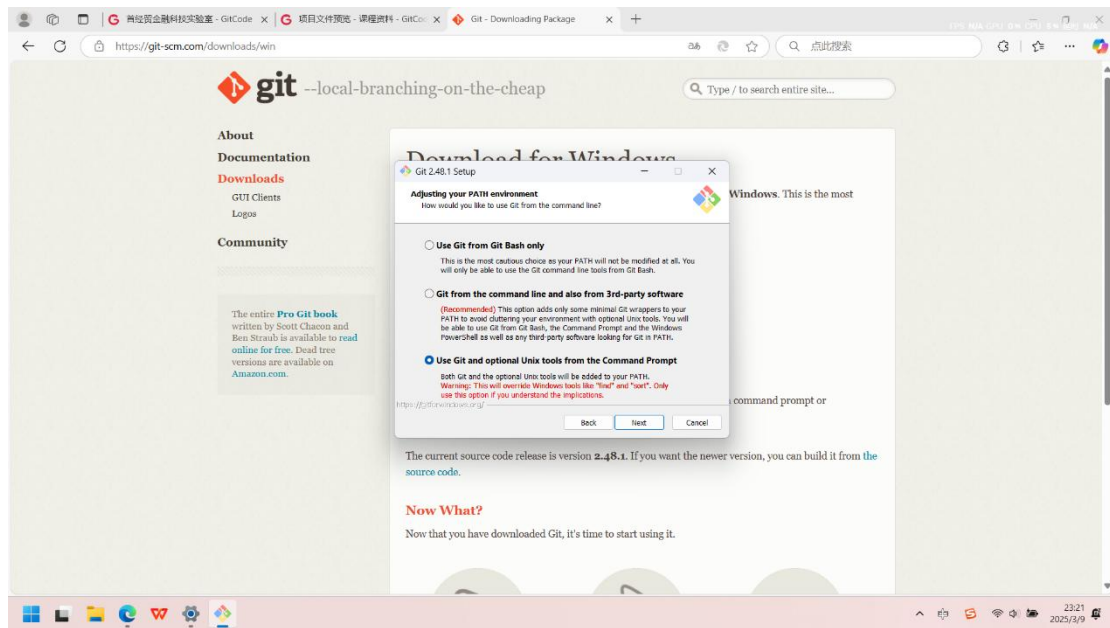
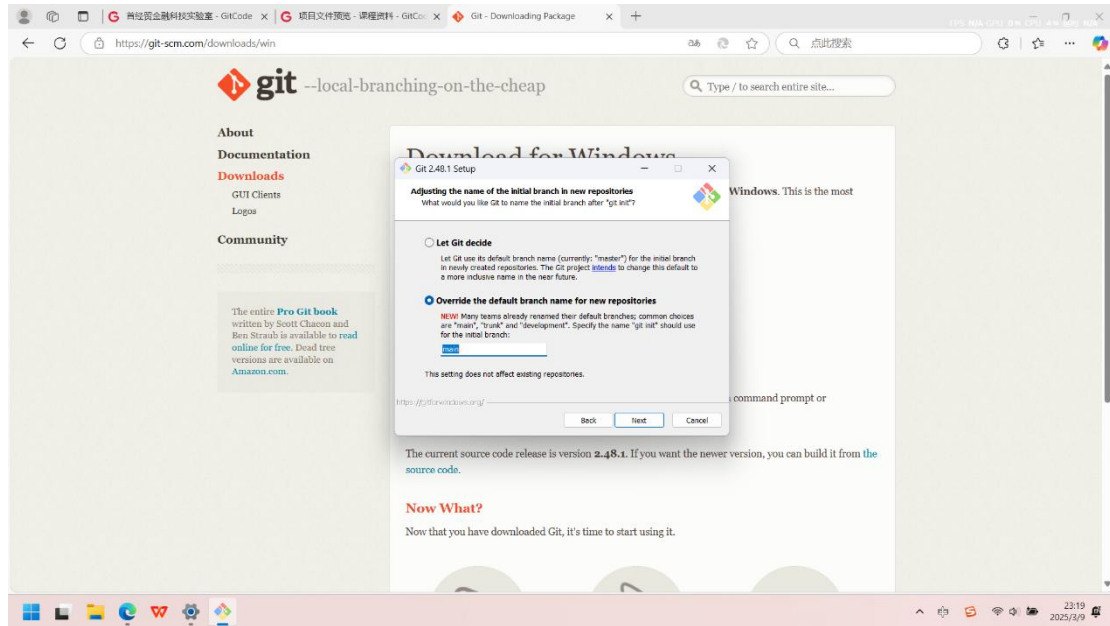


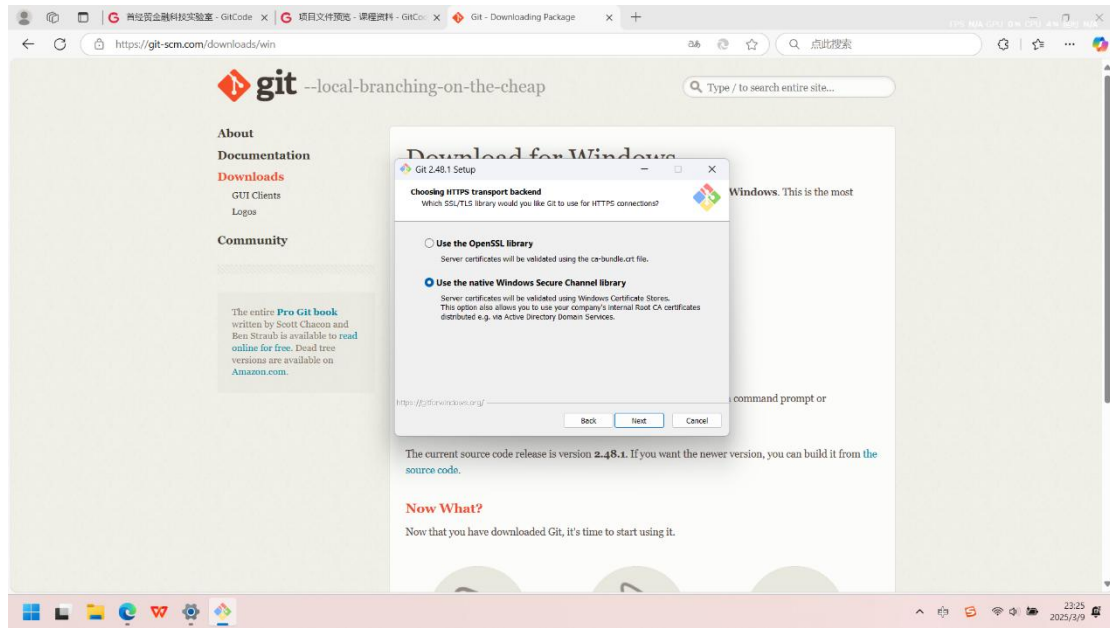
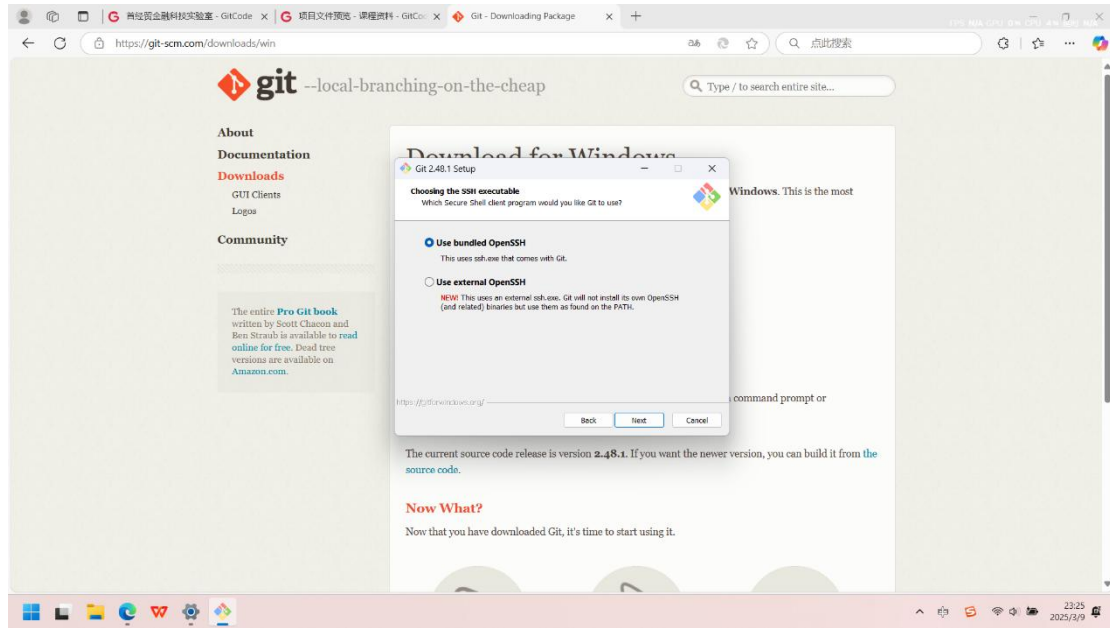
1.安装 VS Code（代码编辑器）

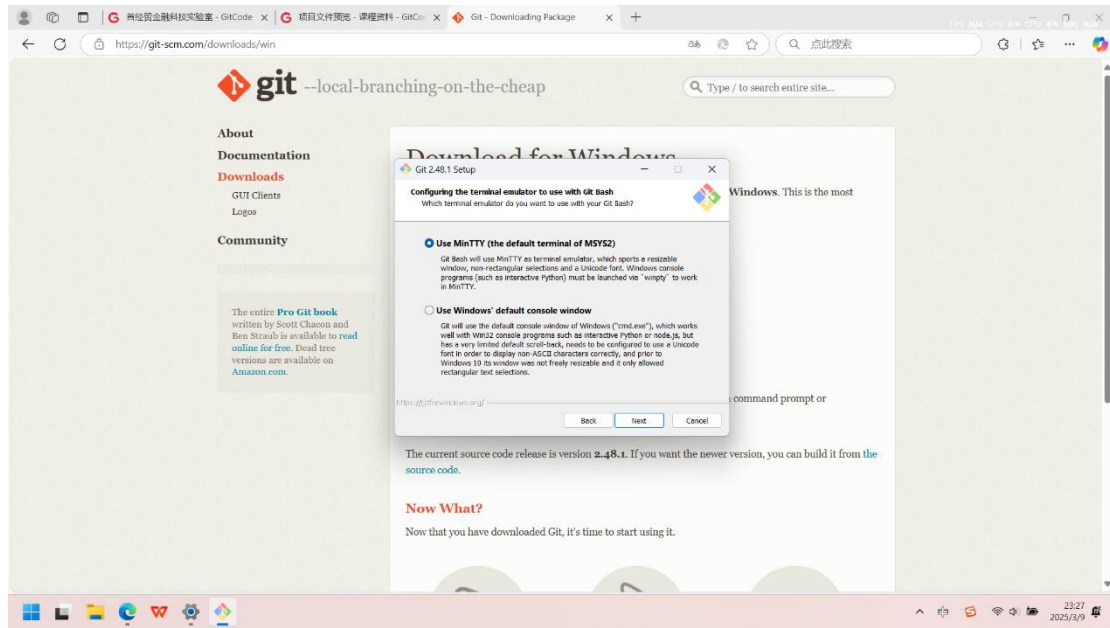
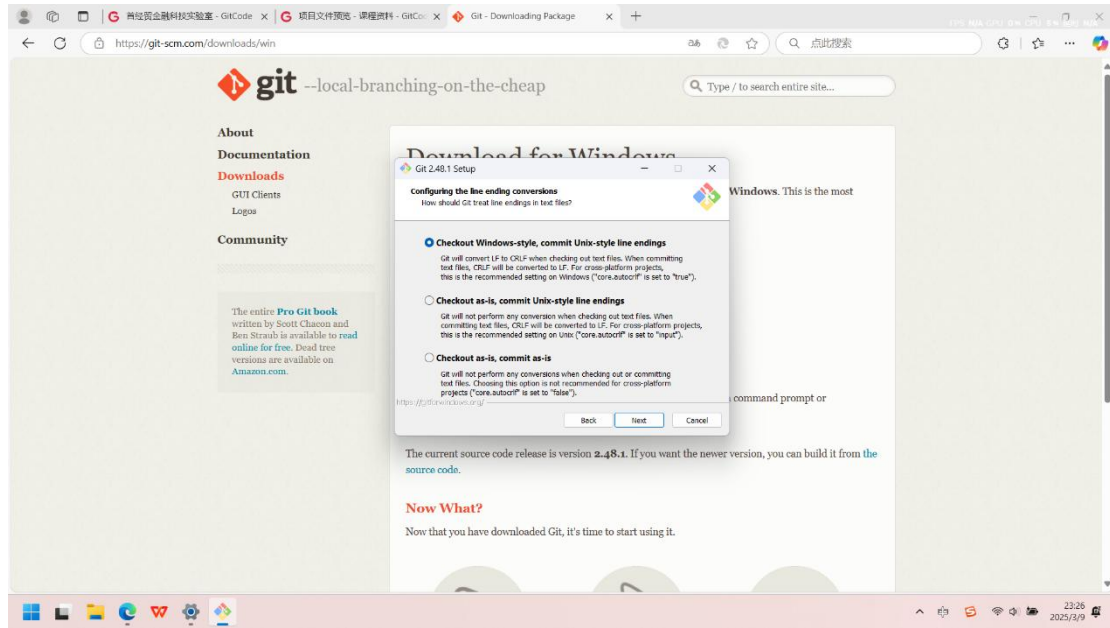
在老师精心录制的视频教学指导下，我顺利完成 VS Code 安装。下载官网安装包后，按提示一步步操作，轻松完成安装。初次体验，界面简洁，功能栏布局清晰，代码智能提示、语法检查等功能实用，极大提升了编程效率，个人认为它是一款便捷强大的编程工具。

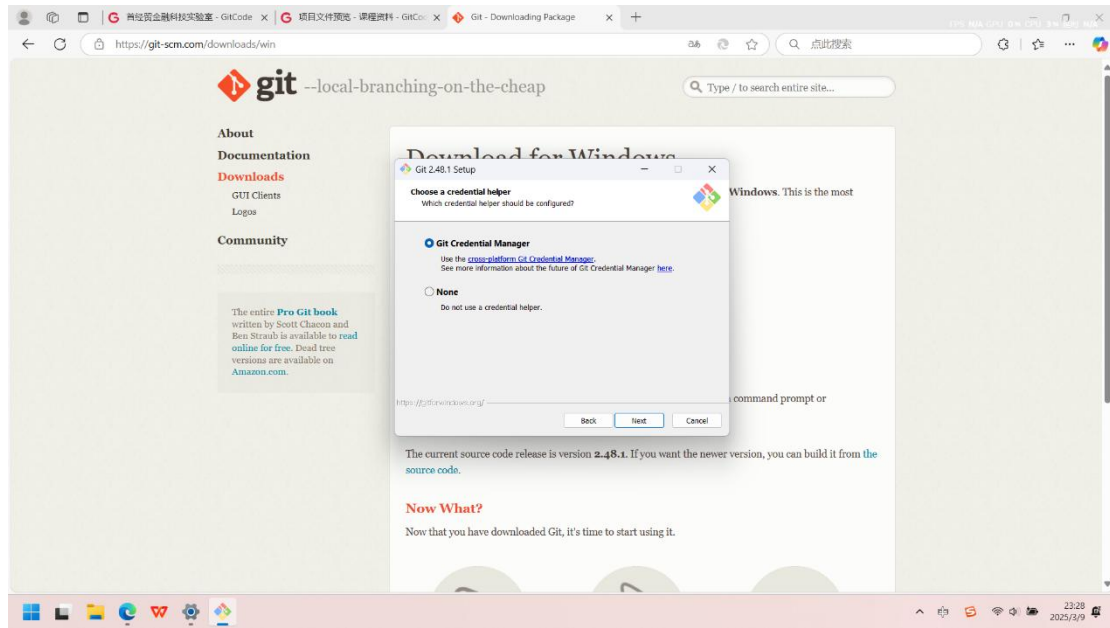
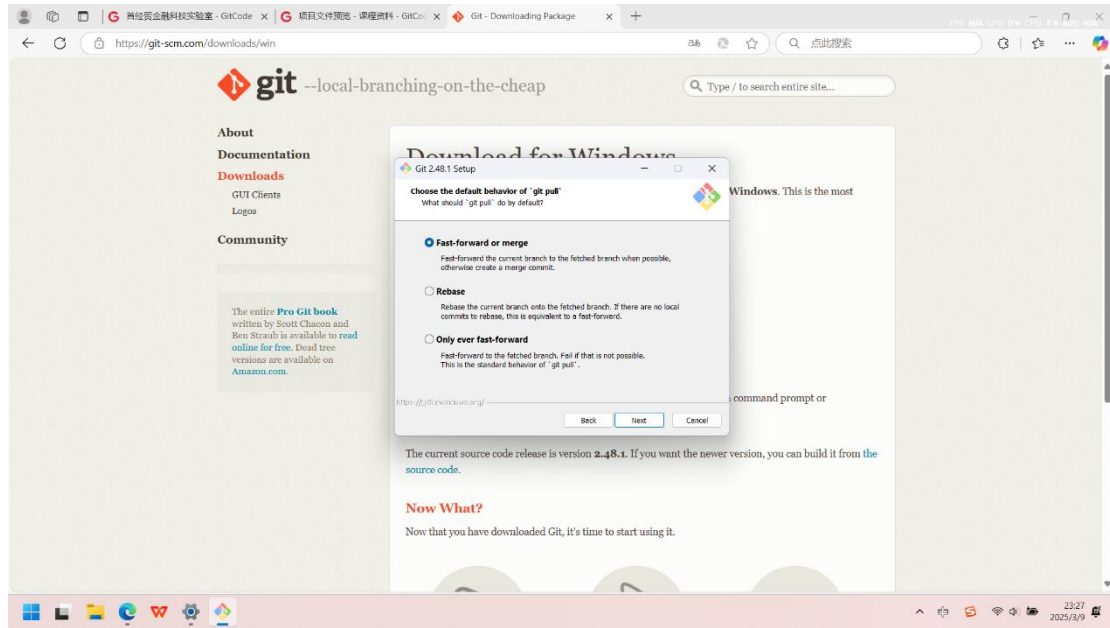


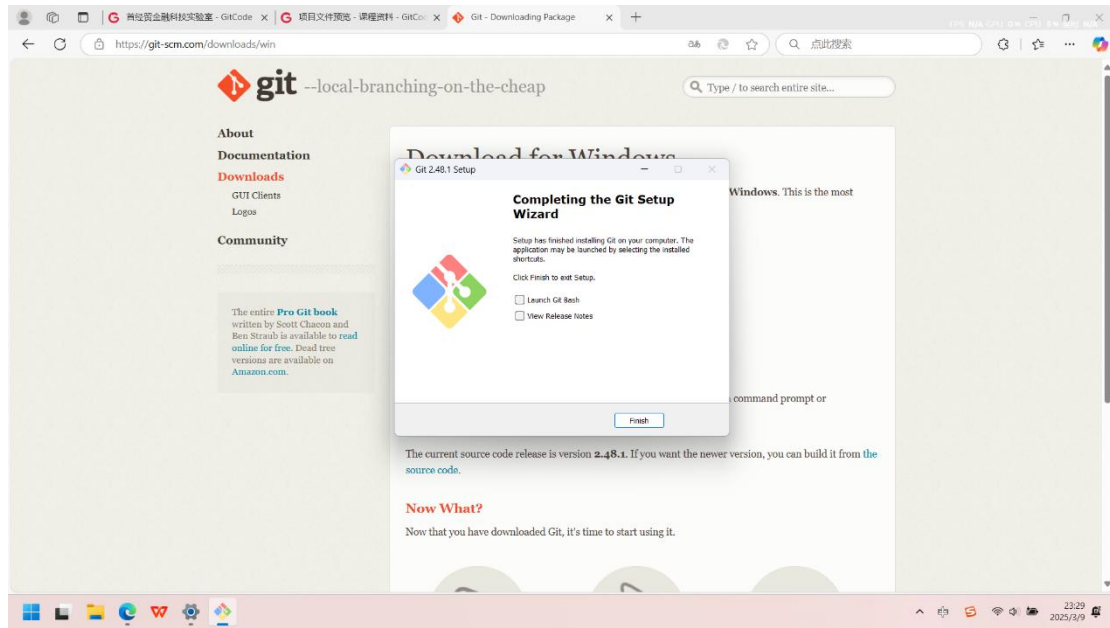
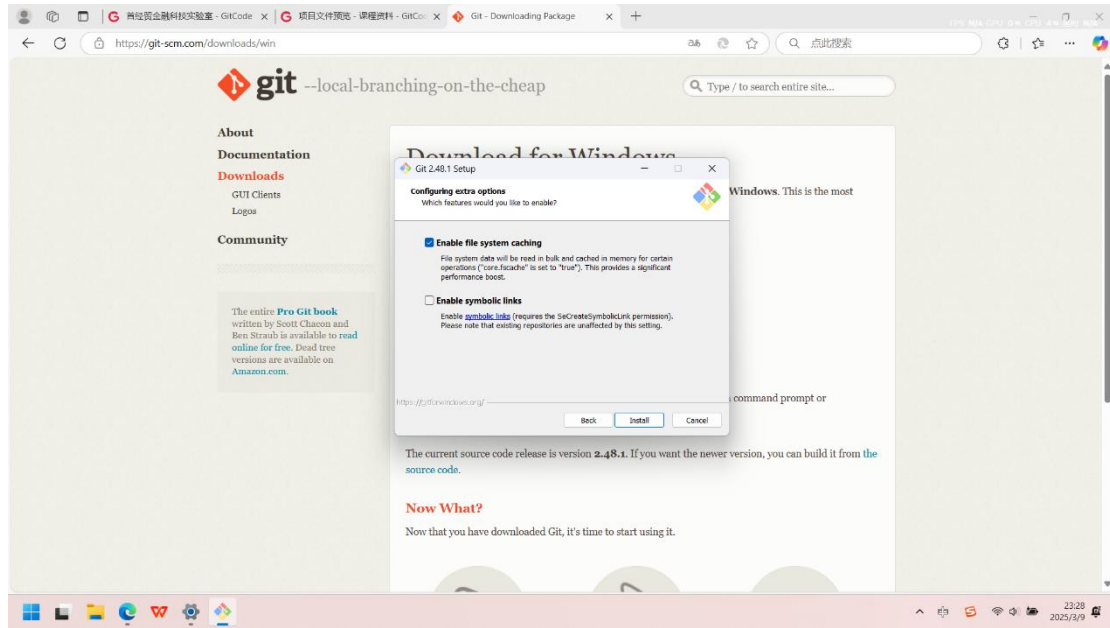


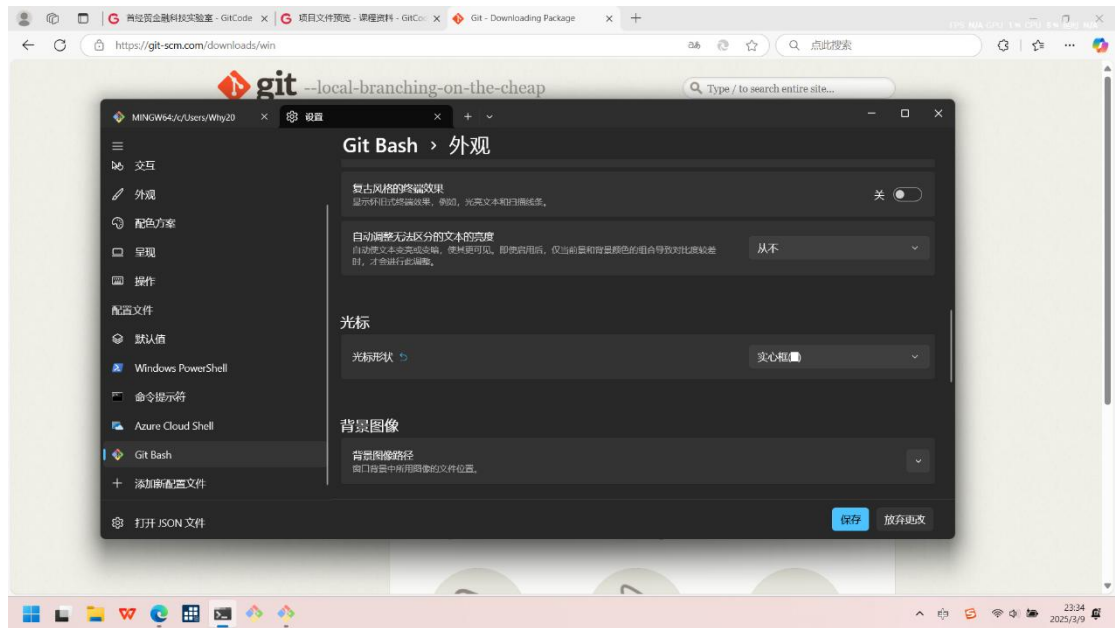






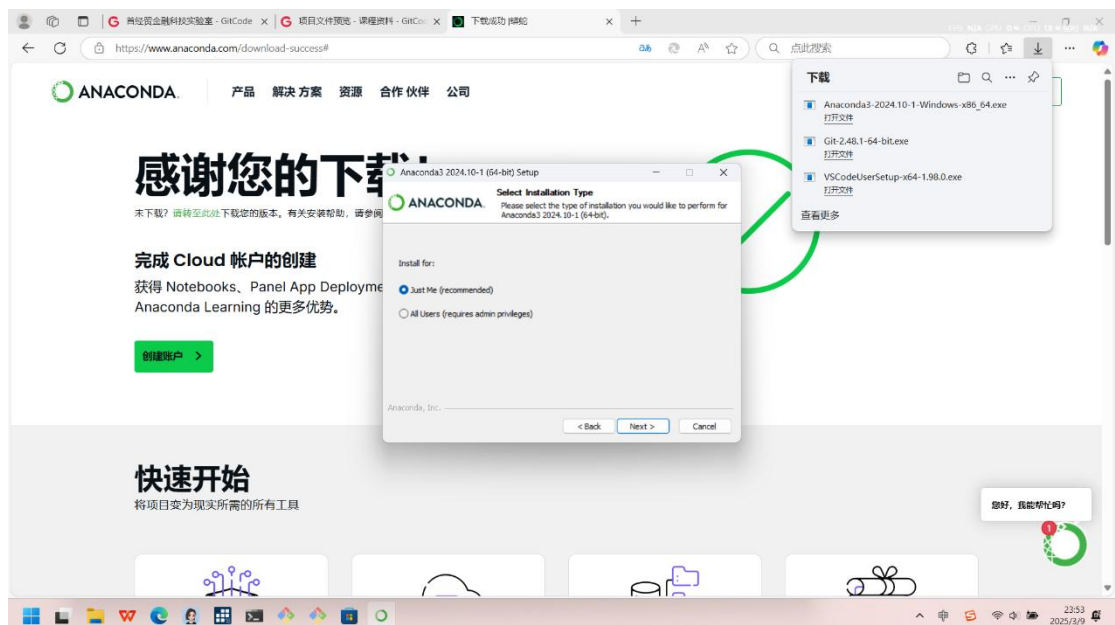


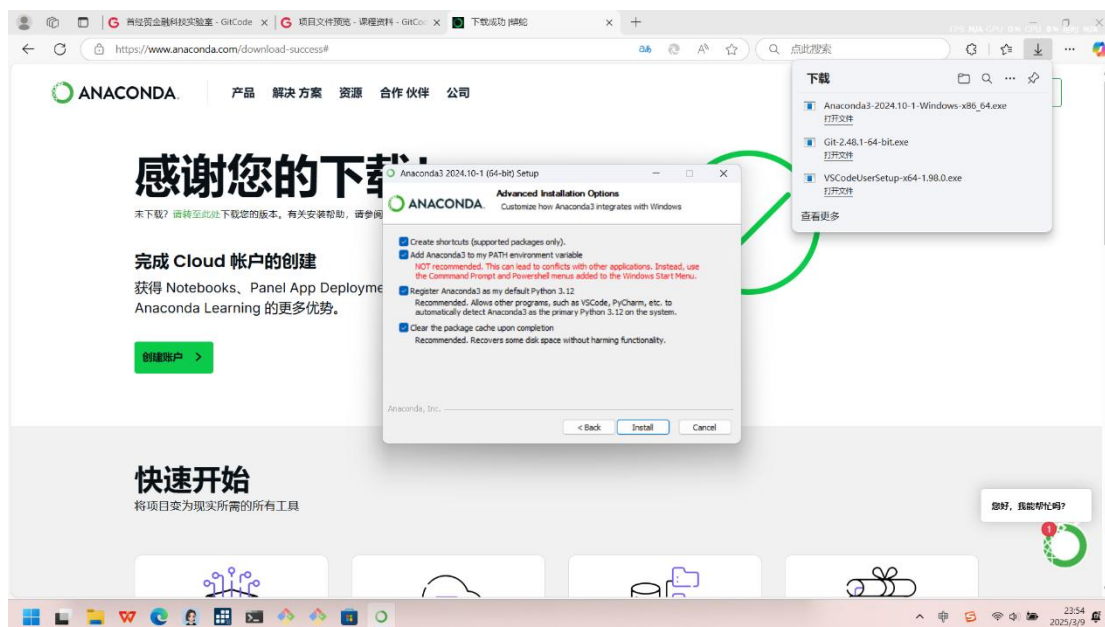
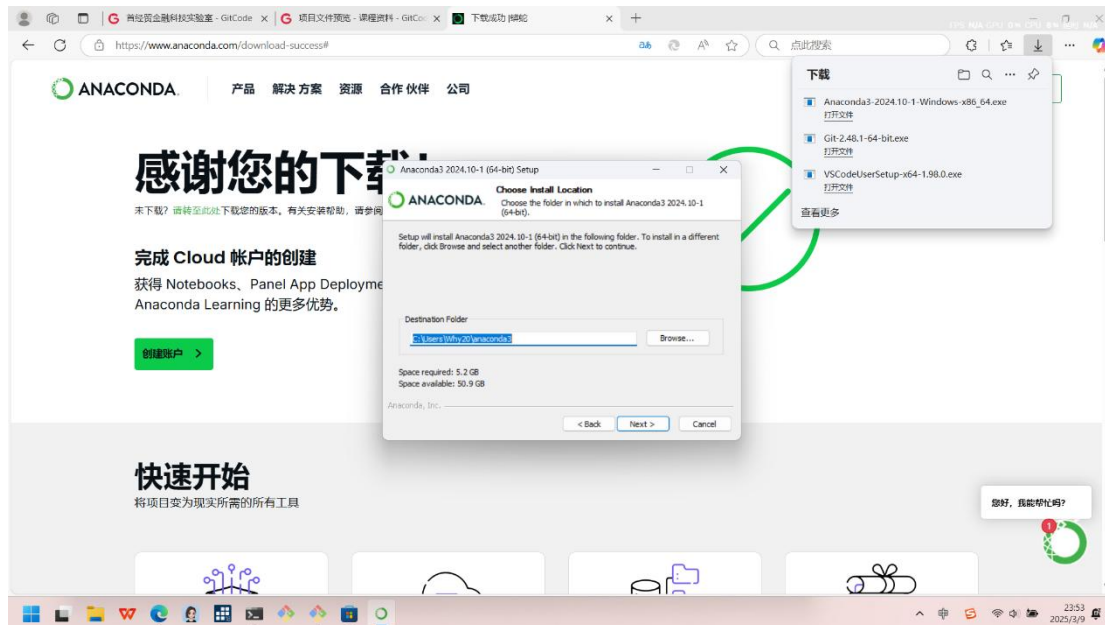




2. 安装 Git（代码版本管理软件）

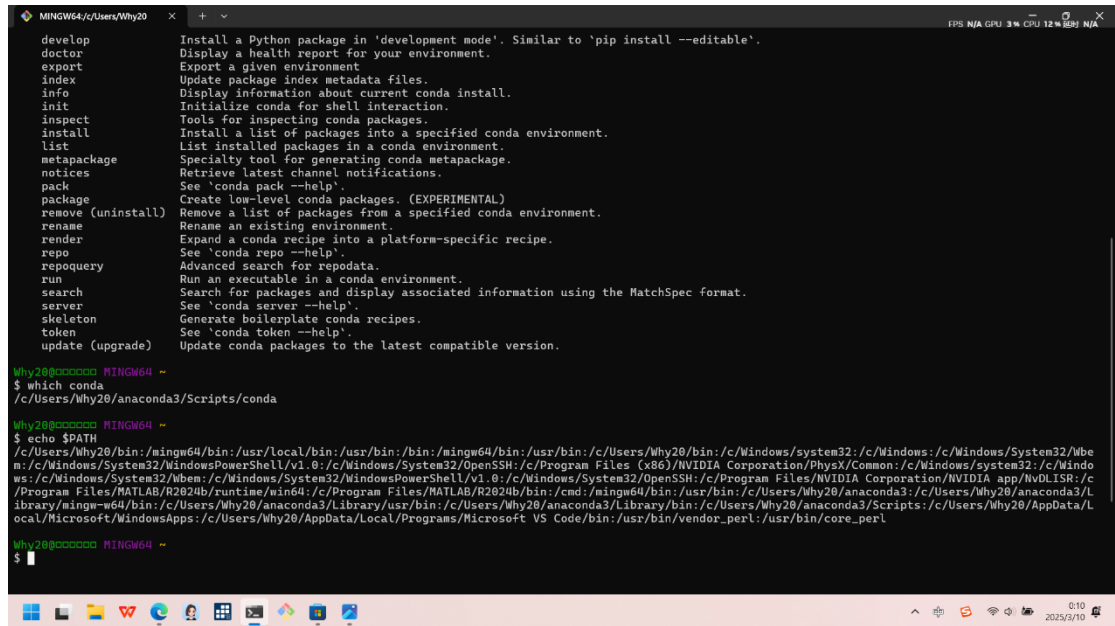
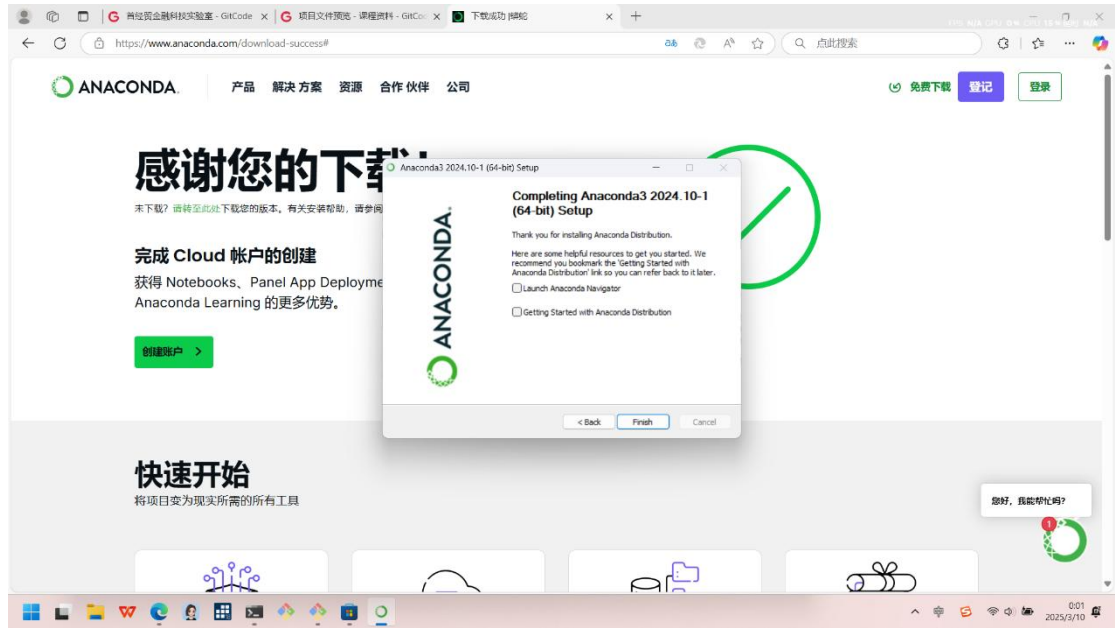
在完成老师布置的 Git 安装任务过程中，我收获颇丰。从官网下载安装程序后，安装向导界面简洁明了，一步步引导操作。过程中，众多用户偏好的个性化设置选项让我印象深刻。可自定义文本编辑器，根据自身习惯选择换行符格式，还能按需配置路径环境变量。这些丰富且细致的设置，充分展现了 Git 对不同用户需求的尊重，极大提升了用户自主性。安装完成后，使用起来顺畅便捷，整个体验十分人性化，让我切实感受到它作为高效版本控制工具的魅力。

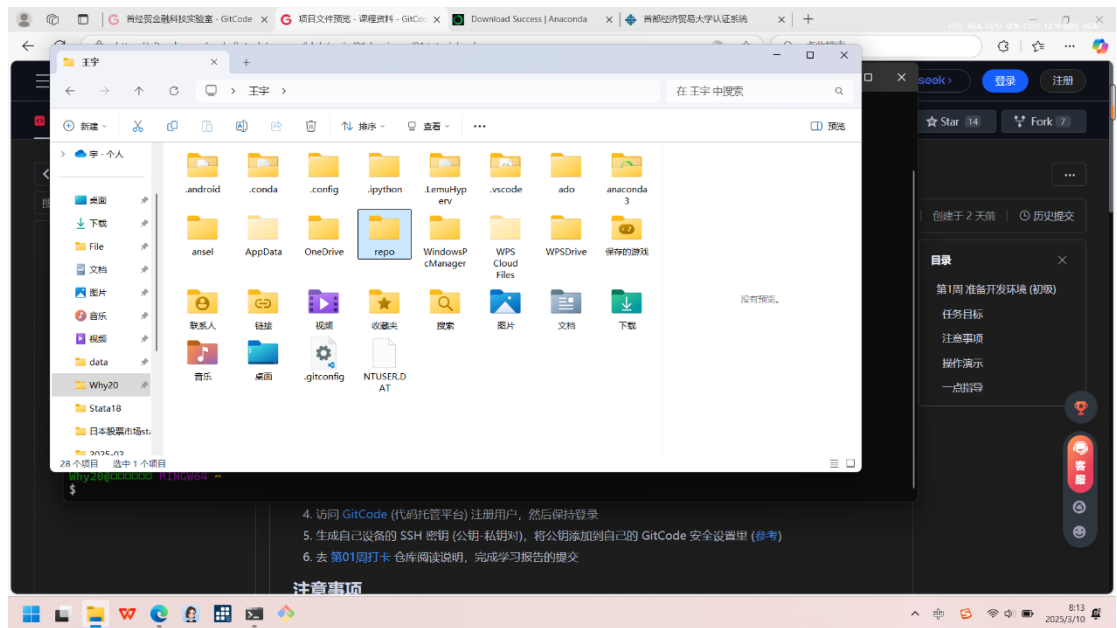
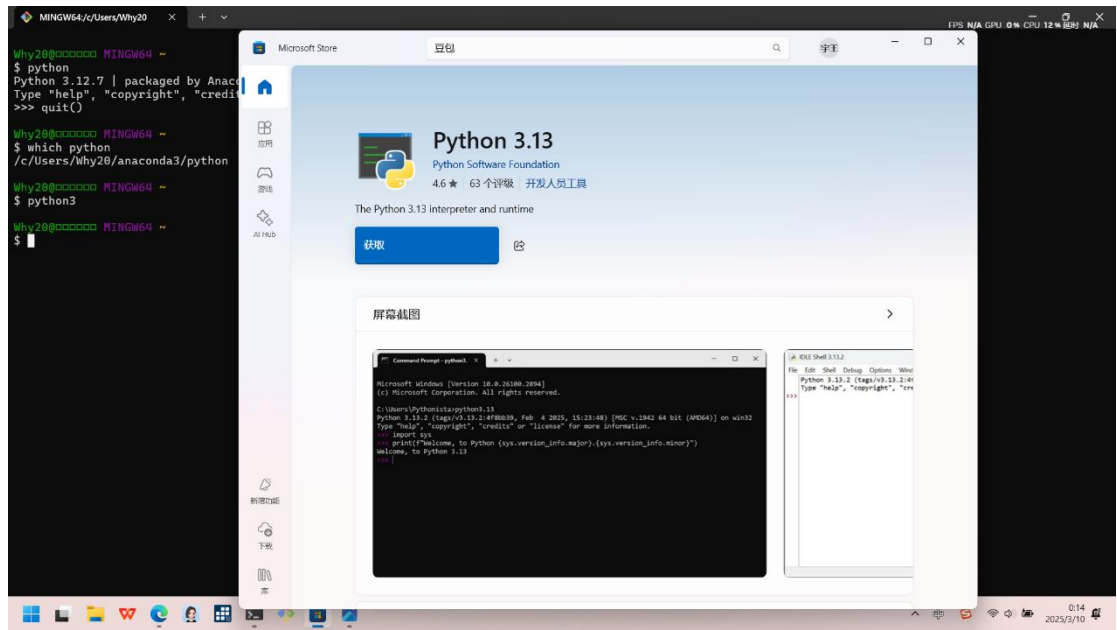


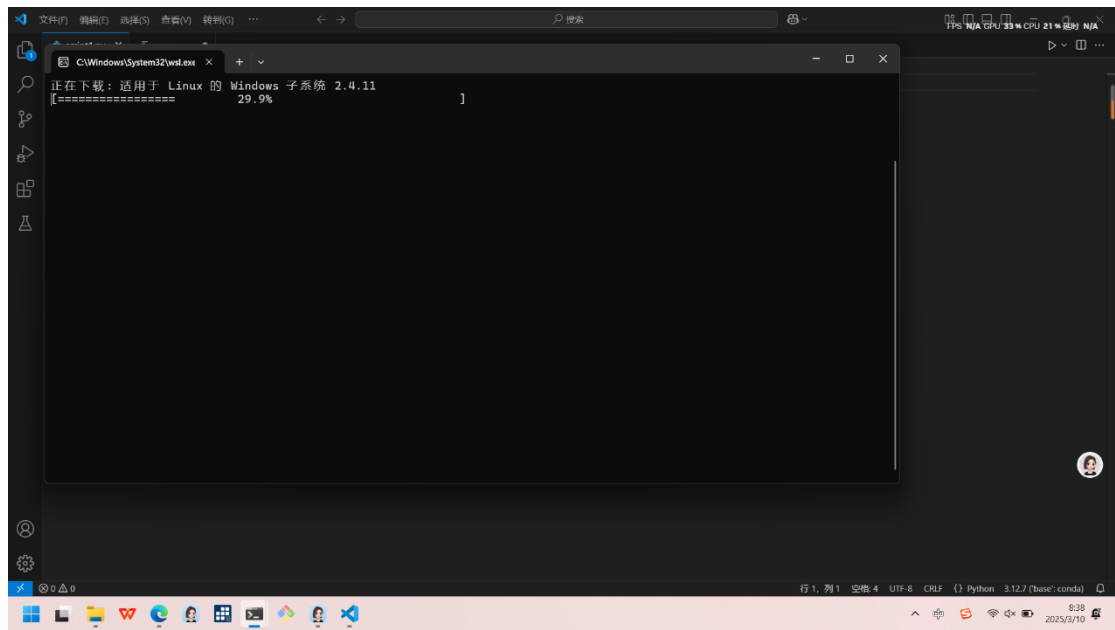
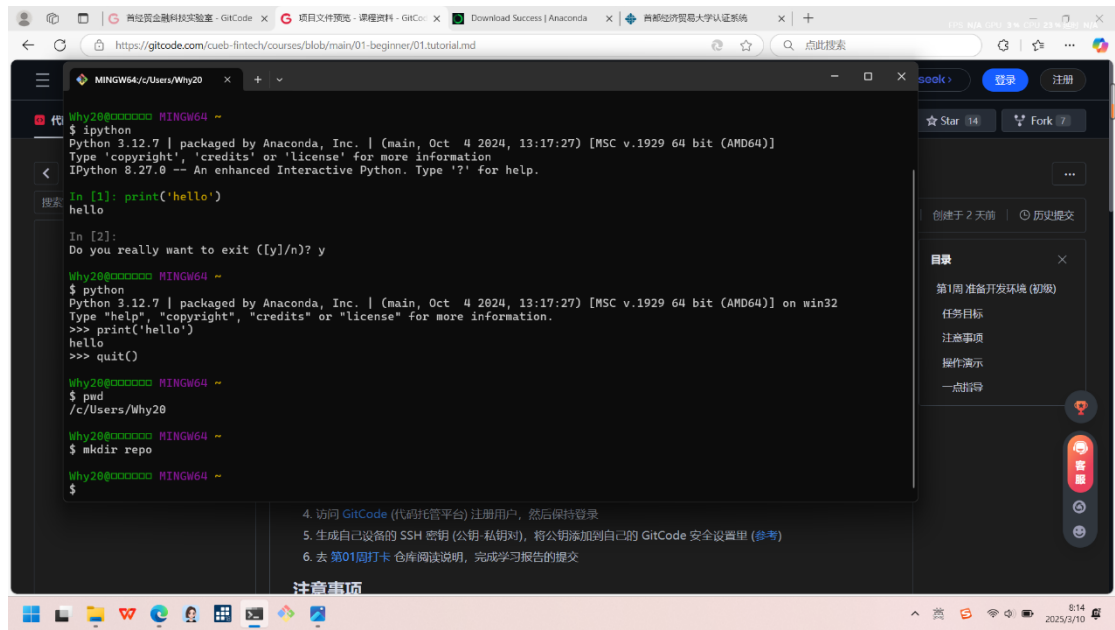


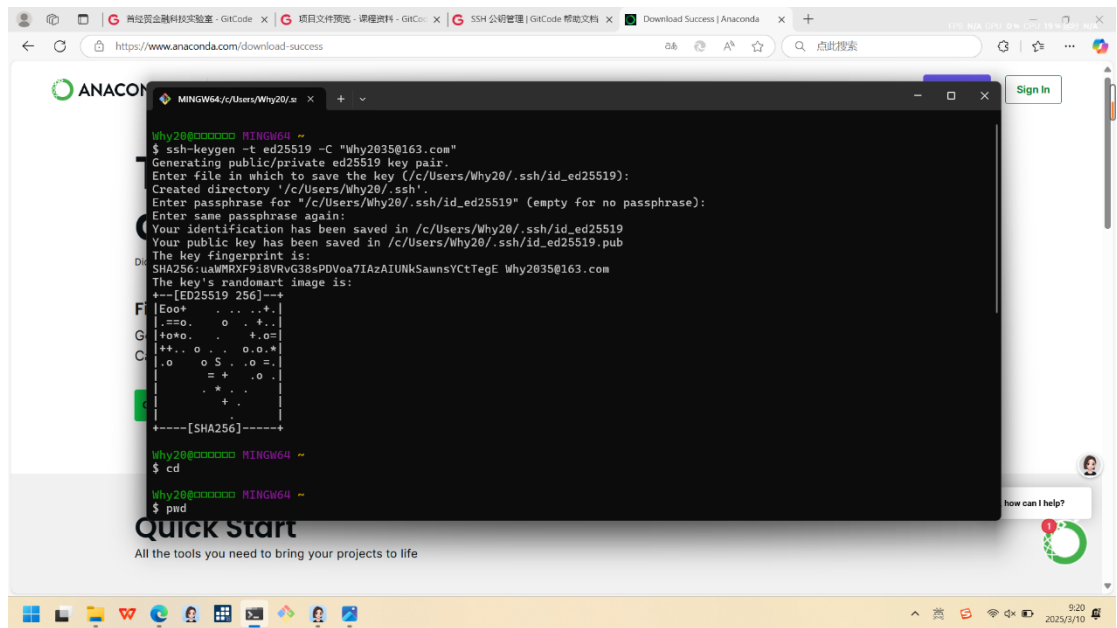
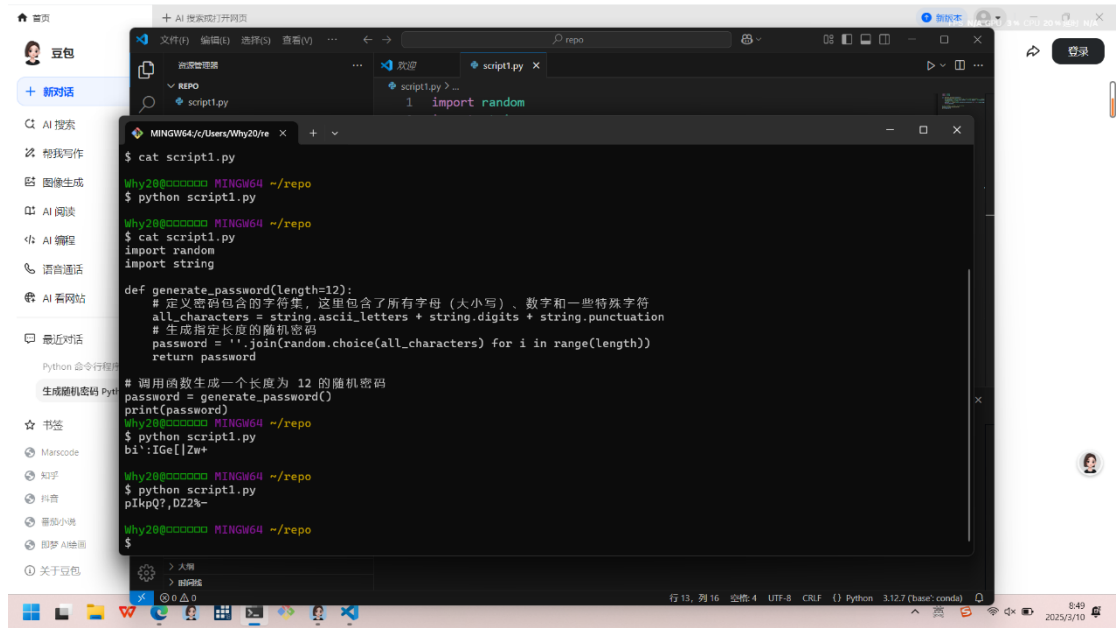
3.安装 Anaconda（Python 解释器）

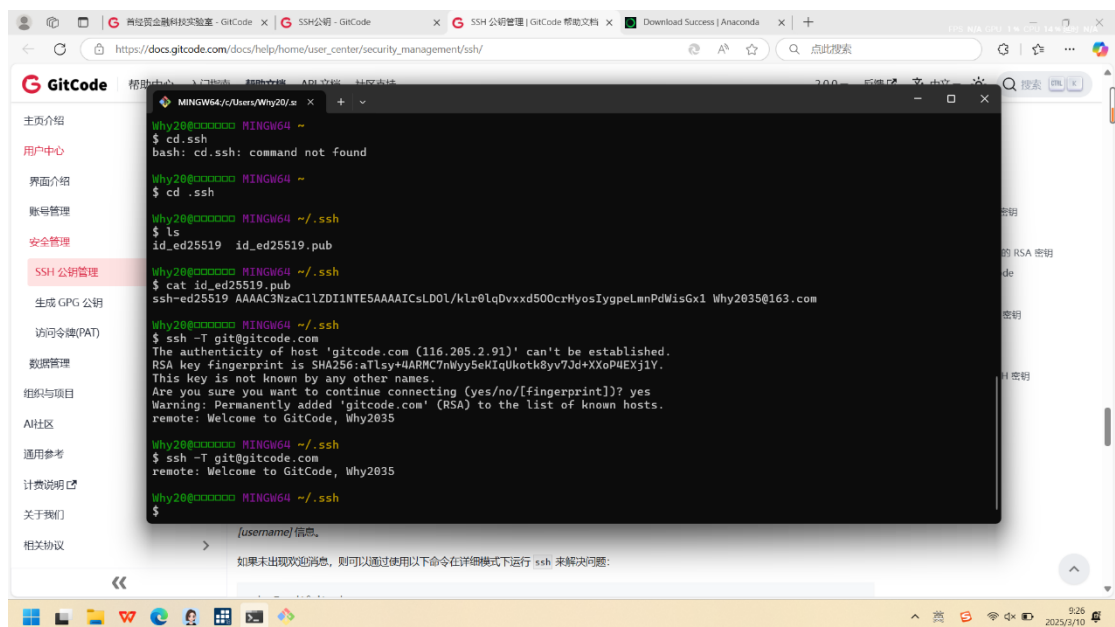
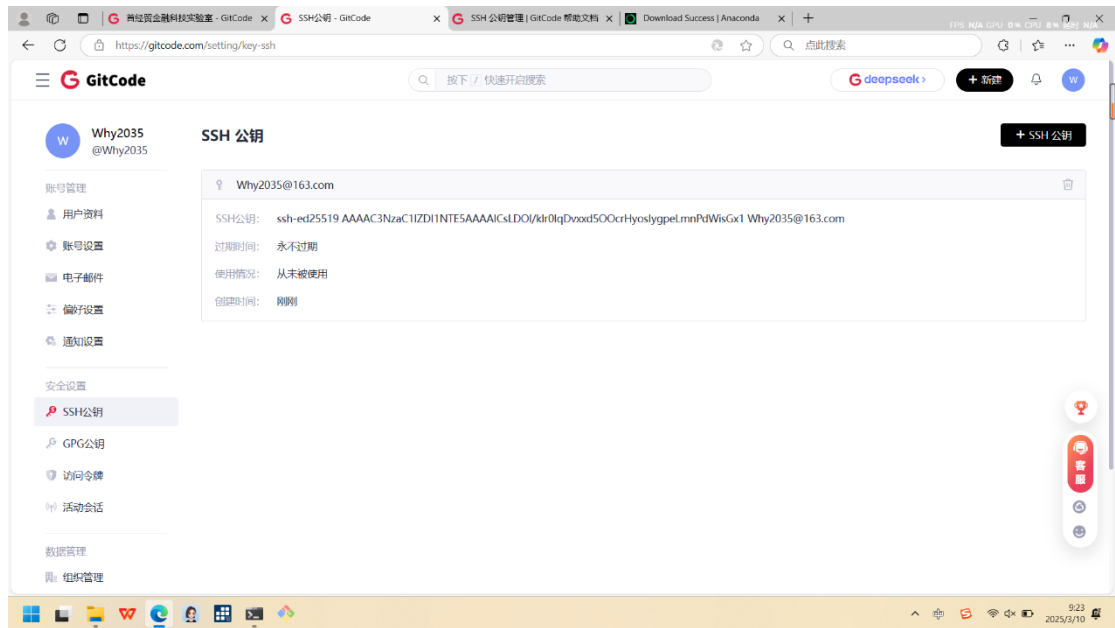
从官网下载对应版本安装包后，我开始进行安装。安装向导以清晰易懂的界面，有条不紊地引导我完成每一步。过程中，有大量个性化设置选项，例如可自主选择安装路径，决定是否将 Anaconda 添加到系统环境变量中，还能根据需求选择安装的 Python 版本。这些丰富的设置充分考量了不同用户的使用习惯和需求，极大提升了安装的灵活性。安装完成后，经初步使用，Anaconda 强大的功能与便捷的操作，让我切实体会到其用户友好性，为后续学习奠定良好基础。











4. 访问 GitCode (代码托管平台) 注册用户，然后保持登录
5. 生成自己设备的 SSH 密钥 (公钥-私钥对)，将公钥添加到自己的 GitCode 安全设置里

在操作中，我生成自己设备的 SSH 密钥（公钥-私钥对），并将公钥添加到 GitCode 安全设置里面。起初，面对指令和步骤，我有些不知所措，但随着一步步实践，逐渐掌握了要领。生成密钥时，我深刻体会到非对称加密技术的优越性。私钥如同保险柜的钥匙，必须妥善保管；公钥则可放心分享。将公钥添加到 GitCode，就像给仓库加上了坚固的锁。这一过程不仅让我掌握了新技能，更让我切实感受到网络安全的重要性，密钥技术为数据传输与存储筑牢了坚实防线。

Why2035/第01周打卡

fork 源: 普经资金融科技实验室 / 第01周打卡

代码 Issues Pull Requests 讨论 项目成员 1 分析 项目设置

main 分支 1 Tags 0 搜索文件 + 下载zip Clone

文件	最后提交记录	最后更新时间
.gitignore	Initial commit	13 天前
LICENSE	Initial commit	13 天前
README.md	添加了提交 PDF 学习报告的操作步骤	5 天前

提交 PDF 学习报告

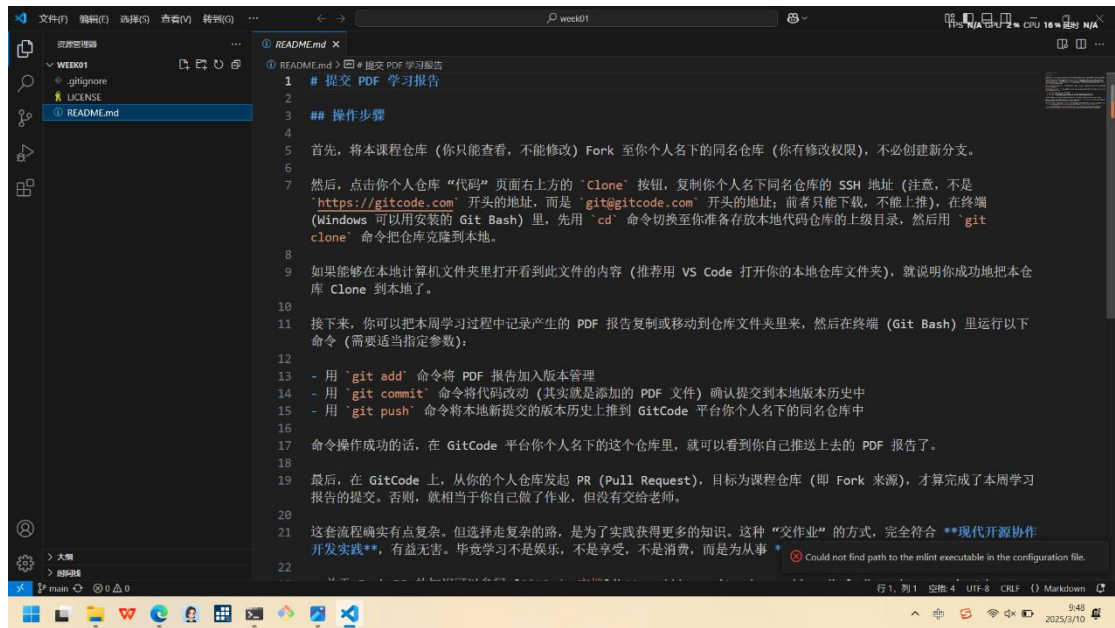
操作步骤

首先, 将本课程仓库 (你只能查看, 不能修改) Fork 至你个人名下的同名仓库 (你有修改权限), 不必创建新分支。

然后, 点击你个人仓库 “代码” 页面右上方的 Clone 按钮, 复制你个人名下同名仓库的 SSH 地址 (注意, 不是 `https://github.com`)

项目介绍
学生 Fork 此仓库并通过 PR 提交第 1 周学习报告
2 提交数
README
规则集
下载使用量
0
项目代码总下载次数 (含Clone, Pull及 zip 包下载), 每周更新
发行板
暂无数据
创建发行版

```
git config --global user.email Why2035@noreply.gitcode.com
Why20@MINGW64: ~/.ssh
$ git config --list --global
core.editor=C:\Users\Why20\AppData\Local\Programs\Microsoft VS Code\bin\code --wait
user.name=Why2035
user.email=Why2035@noreply.gitcode.com
Why20@MINGW64: ~/.ssh
$ pwd
/c/Users/Why20/.ssh
Why20@MINGW64: ~/.ssh
$ cd repo
bash: cd: repo: No such file or directory
Why20@MINGW64: ~/.ssh
$ pwd
/c/Users/Why20/.ssh
Why20@MINGW64: ~/.ssh
$ cd
Why20@MINGW64: ~
$ pwd
/c/Users/Why20
Why20@MINGW64: ~
$ cd repo
Why20@MINGW64: ~/repo
$ pwd
/c/Users/Why20/repo
Why20@MINGW64: ~/repo
$ git clone git@gitcode.com:Why2035/week01.git
Cloning into 'week01'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
```



6.通过 fork、Git 里面的 clone、add、commit 以及 push 命令完成 PDF 版的第一周学习报告在个人仓库上的提交，最终在 GitCode 上，从个人仓库发起 PR，目标设置为课程仓库（即 fork 来源），完成第一周作业的提交。

本周按照老师要求完成了课程学习报告在 GitCode 上的提交操作。

首先，我将课程仓库 Fork 到个人名下，获得可修改的同名仓库。接着复制个人仓库的 SSH 地址，在终端使用 `git clone` 命令将仓库克隆到本地，成功在本地文件夹中看到文件内容。

随后，我把本周学习的 PDF 报告放入仓库文件夹，在终端依次运行 `git add`、`git commit` 和 `git push` 命令，将报告推送到个人仓库。

最后，在 GitCode 上从个人仓库发起 PR 到课程仓库，完成报告提交。通过这次操作，我对 Git 版本控制和协作流程有了更深入的理解，也体会到了其在项目管理中的高效与便捷。