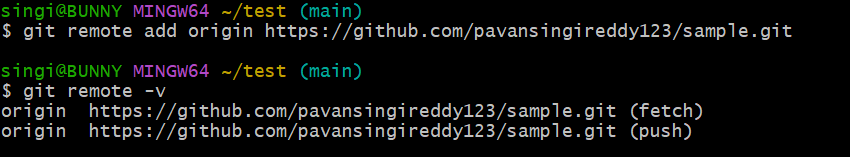
1. **Collaboration:**

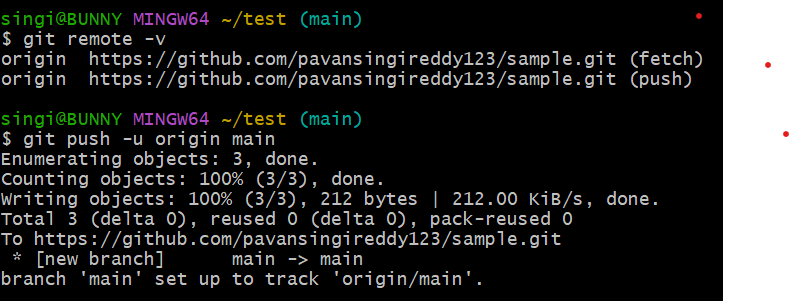
* **git remote:** Adds a new remote repository. This is useful when you want to link your local repository with a remote one, like GitHub.

**git remote add <name><url>**

****

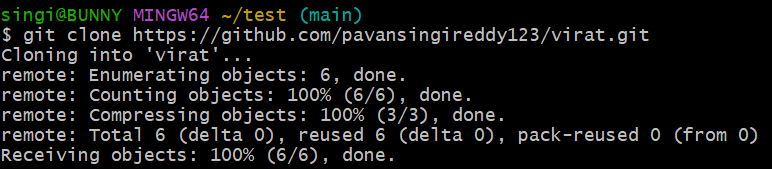
* **git push:** Used to transfer the commits or pushing the content from the local repository to the remote repository. The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

**git push –u origin <master|current directory>**

****

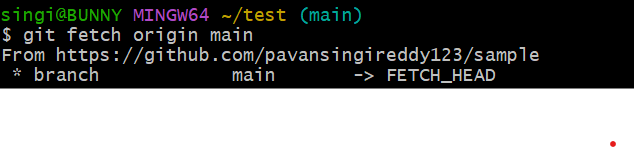
* **git clone:** used to create a local working copy of an existing remote repository.The command downloads the remote repository to the local computer.

**gitclone <remote url>**

****

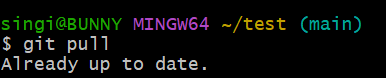
* **git fetch :** Downloads changes from a remote repository without applying them to your working sdirectory. You can later merge these changes.

**git fetch <remote>**



* **git pull:** Fetches and integrates changes from a remote repository by rebasing instead of merging. This creates a linear history.

**git pull <remote><branch>**



**Using SSH Key for Authentication in Git-Github:**

Using SSH keys in GitHub allows you to securely authenticate your GitHub account from your computer without needing to repeatedly enter your username and password. This method is more secure and convenient, especially for frequent interactions with GitHub repositories.

**Step 1: Set Up SSH**

**1.1 Check for Existing SSH Keys:**

->Open terminal and run:

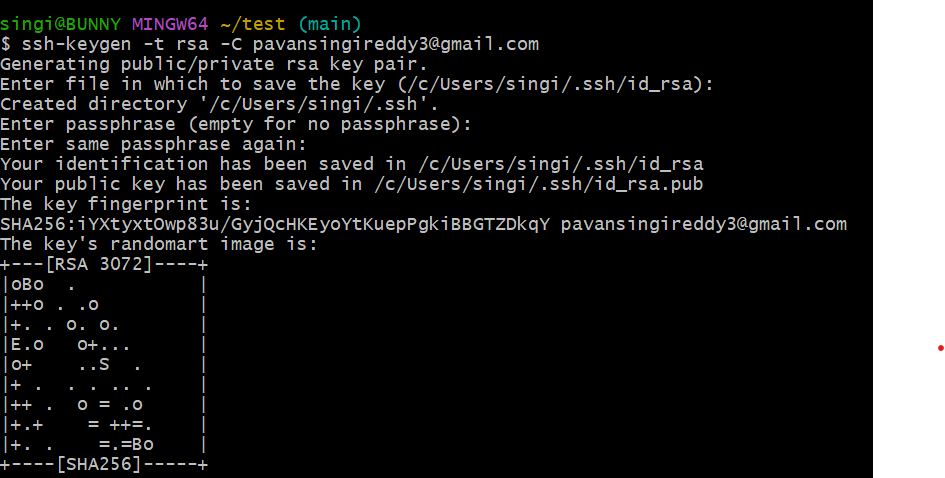
ls -al ~/.ssh`

->If you see `id\_rsa` and `id\_rsa.pub`, you already have an SSH key.

**1.2 Generate SSH Key (if not already present):**

->Run the following command:

`ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"`



-> Press Enter to accept the default location.

->Optionally, set a passphrase for added security.

->SSH keys are saved in `~/.ssh/id\_rsa` (private key) and `~/.ssh/id\_rsa.pub` (public key).

**Step 2: Add SSH Key to GitHub:**

**2.1 Copy SSH key to clipboard:**

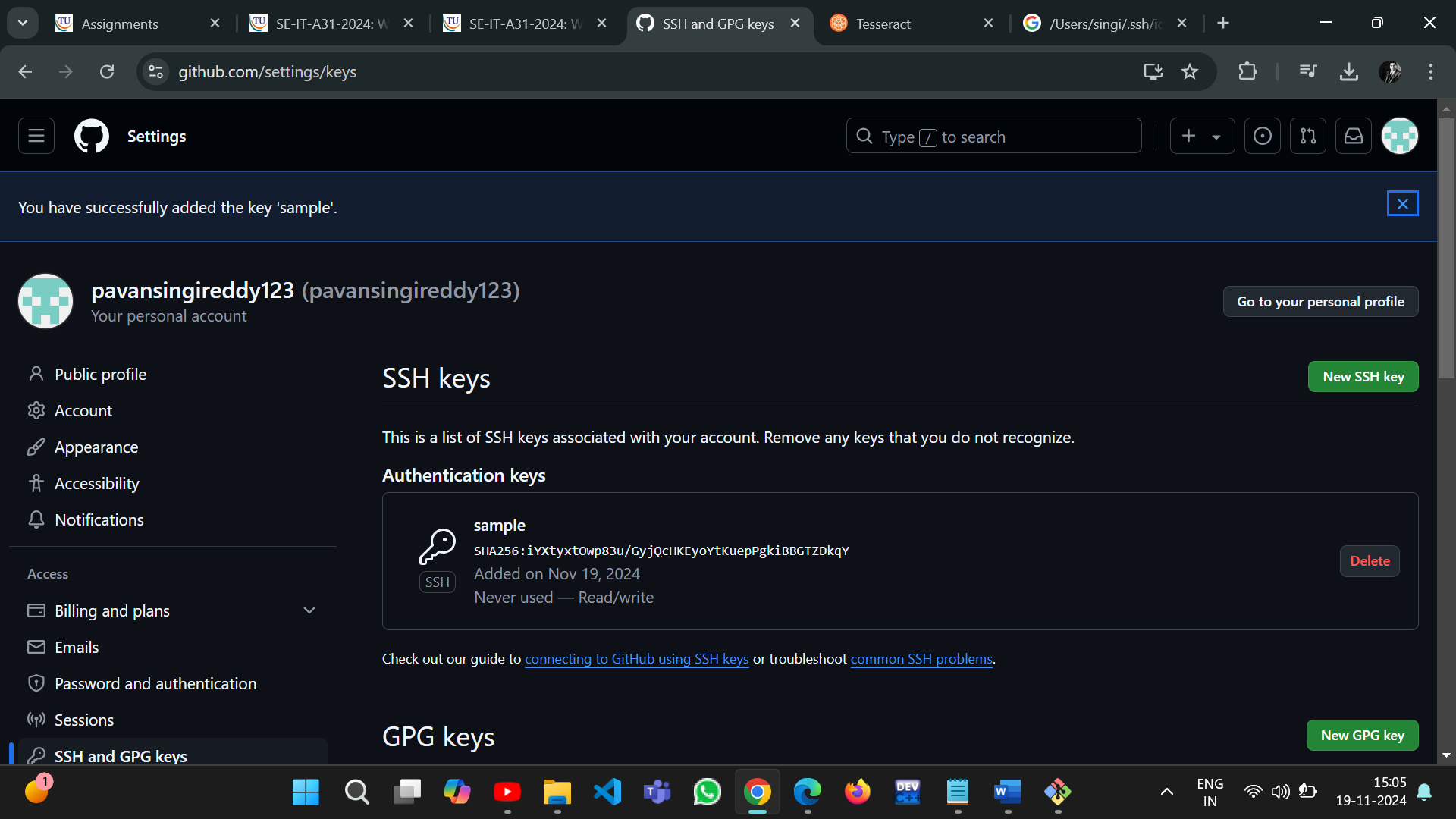
-> Run this command:

`cat ~/.ssh/id\_rsa.pub`

->Copy the entire output.

**2.2 Add SSH Key in GitHub:**

-> Go to GitHub -> Settings -> SSH and GPG keys -> New SSH key.



-> Paste the copied key in the "Key" field and give it a title.

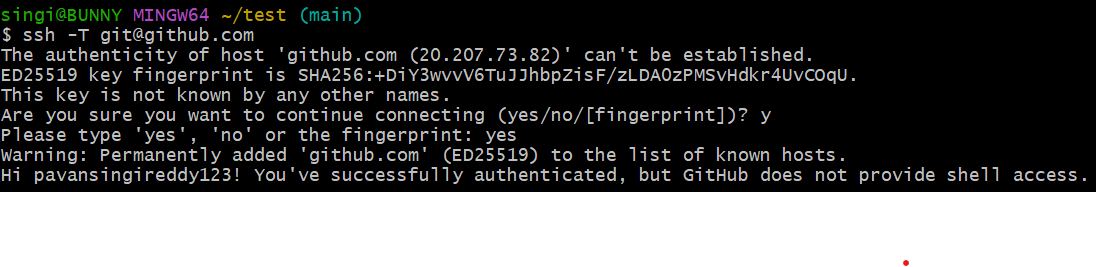
->Click "Add SSH key."

**Step 3: Test SSH Connection:**

3.1 Test the connection:

-> Run the command:

`ssh -T [git@github.com`](mailto:git@github.com%60)



->You should see a message like:

"Hi username! You've successfully authenticated."

**Fork :**

In Git, a **"fork"** refers to a copy of a repository from one user's account to another user's account on a platform like GitHub. Forking is commonly used to contribute to open-source projects or collaborate on projects where you don't have direct write access to the original repository. Forking allows you to work independently.

**Step 1: Fork a Repository**:

-> 1.1 Go to the repository you want to fork (e.g., https://github.com/.../............).

-> 1.2 Click the "Fork" button in the top-right corner to create a copy under your account.

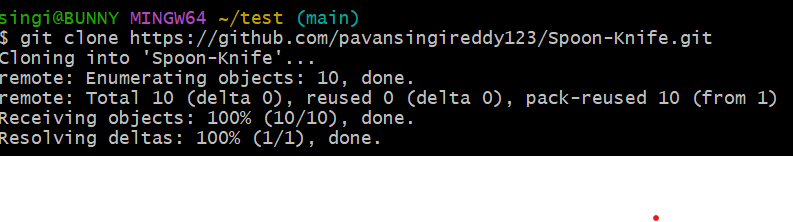
**Step 2: Clone the Forked Repository:**

-> 2.1 Navigate to your forked repository on GitHub.

->2.2 Click the "Code" button and copy the HTTPS or SSH URL.

->2.3 Clone the repository to your local machine:

`git clone [git@github.com:yourusername/Spoon-Knife.git`](mailto:git@github.com:yourusername/Spoon-Knife.git%60)



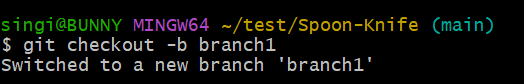
->2.4 Navigate into the repository:

`cd Spoon-Knife`



**Step 3: Make Changes and Push:**

->3.1 Create a New Branch:

`git checkout -b my-feature-branch`

-> 3.2 Make Changes:

->Edit files or create new ones.

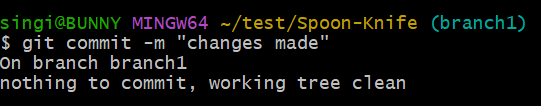
-> Stage changes:

`git add .`



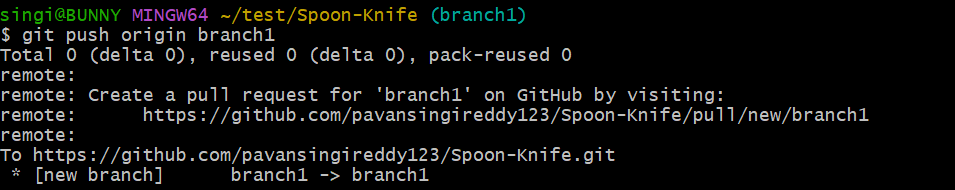
->Commit changes:

`git commit -m "Add my new feature"`



-> 3.3 Push Changes to GitHub:

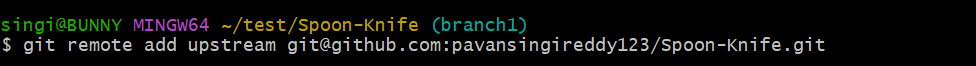
`git push origin my-feature-branch`



**Step 4: Pull from Upstream Repository:**

-> 4.1 Add Upstream Remote:

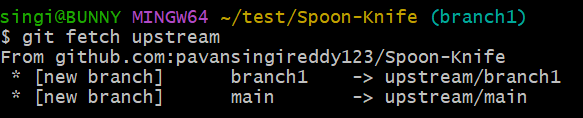
`git remote add upstream [git@github.com:octocat/Spoon-Knife.git`](mailto:git@github.com:octocat/Spoon-Knife.git%60)



-> 4.2 Fetch and Pull Updates:

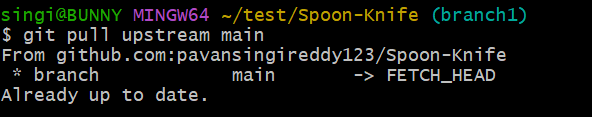
-> Fetch latest changes:

`git fetch upstream`



->Merge changes into your branch:

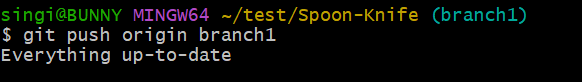
`git pull upstream main`



-> Resolve any conflicts if they arise.

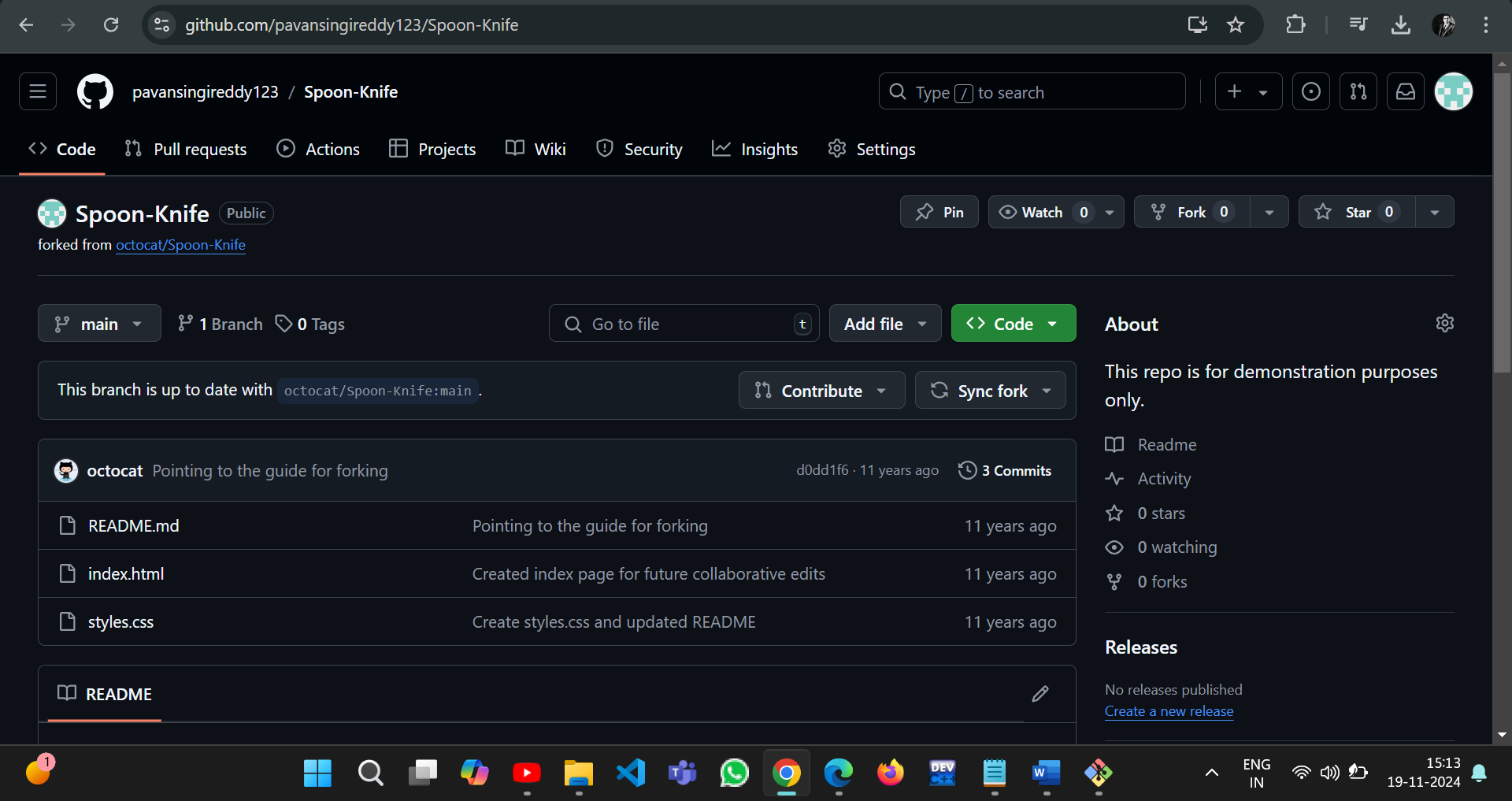
->4.3 Push Merged Changes:

`git push origin my-feature-branch`

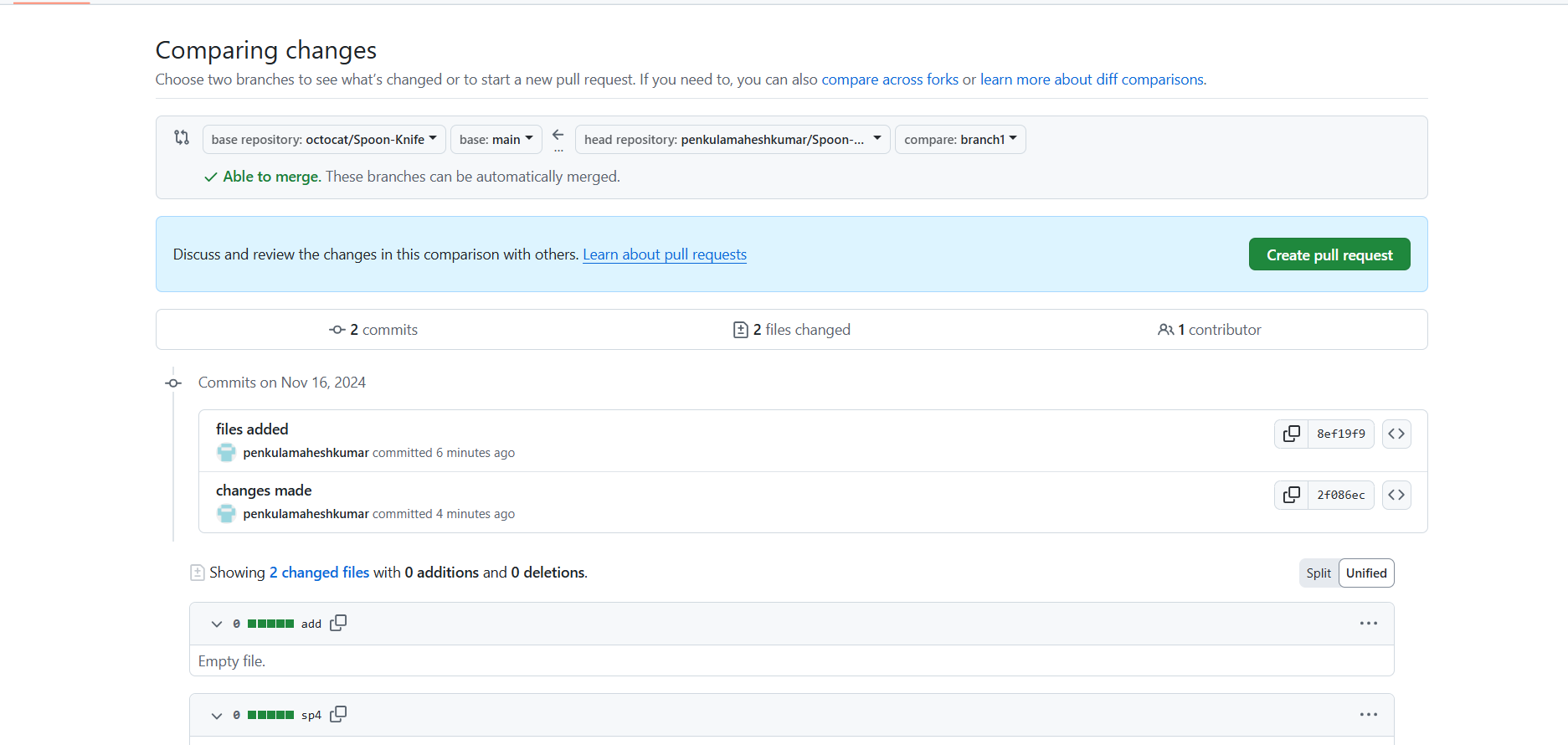


**Step 5: Submit a Pull Request**:

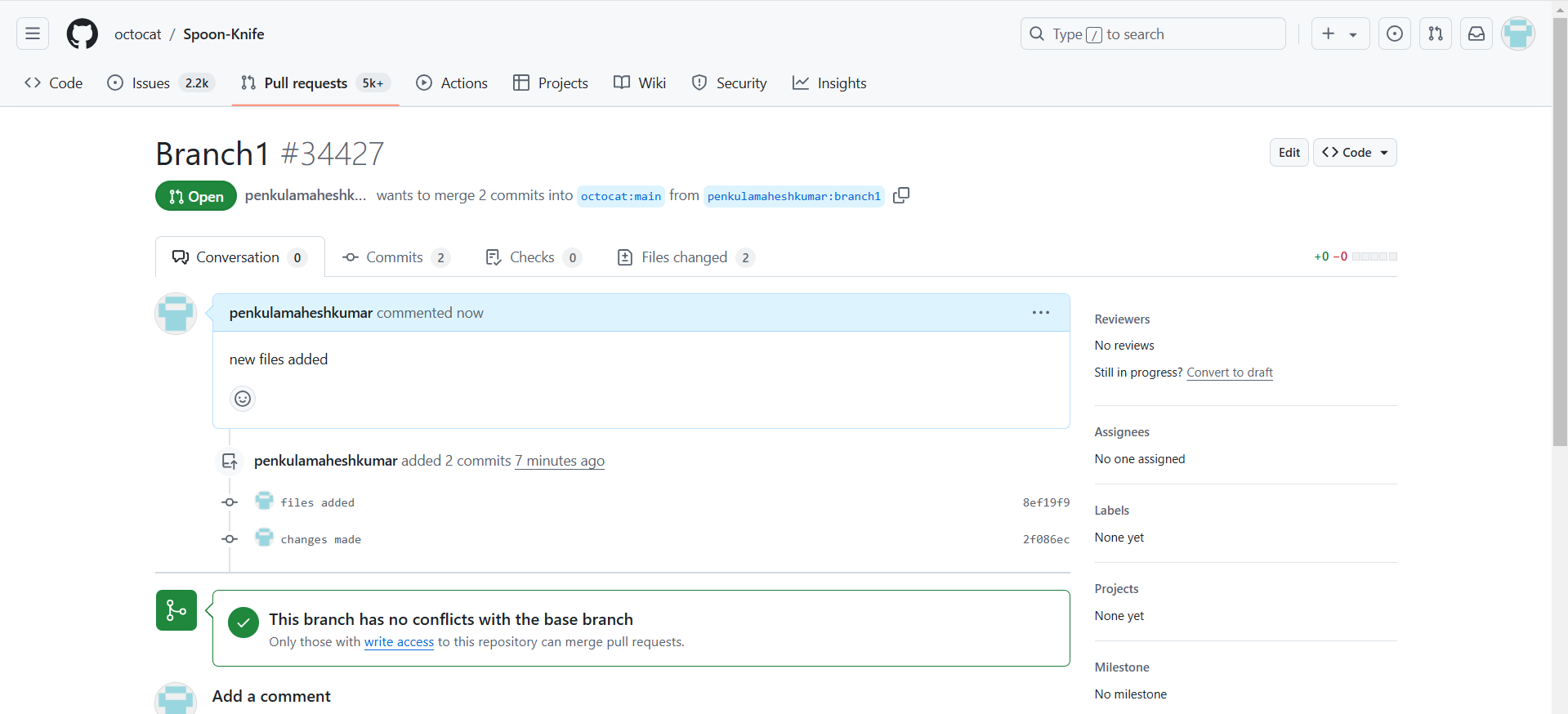
->5.1 Go to your forked repository on GitHub.



-> 5.2 Click the banner suggesting to compare and create a pull request.



-> 5.3 Review your changes and create the pull request to propose changes to the original repository.

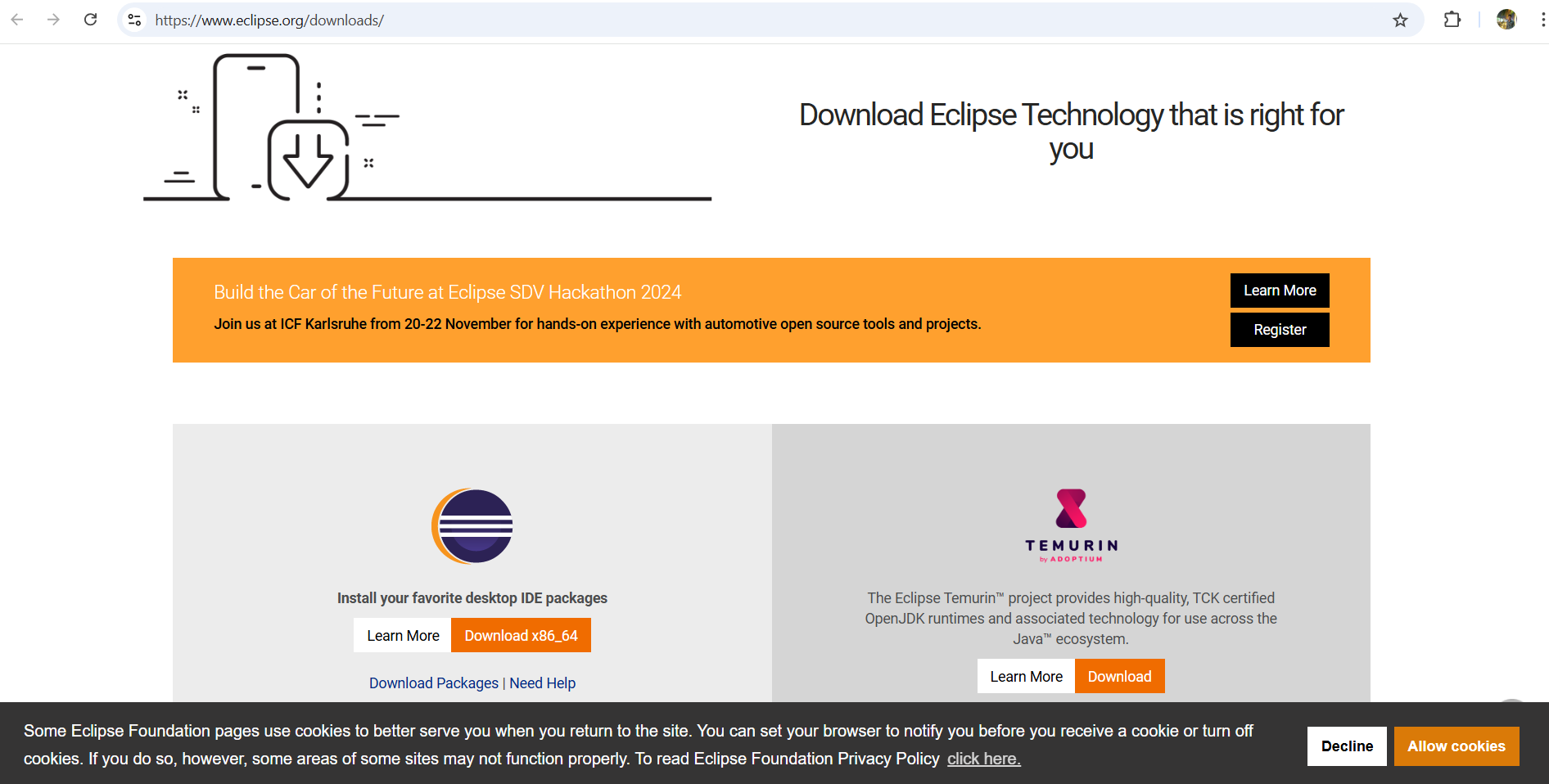


1. **2.1 Installing Eclipse**

To install Eclipse IDE for Enterprise Java Developers (commonly known as Eclipse Enterprise) on your Windows 10 system, follow these steps:

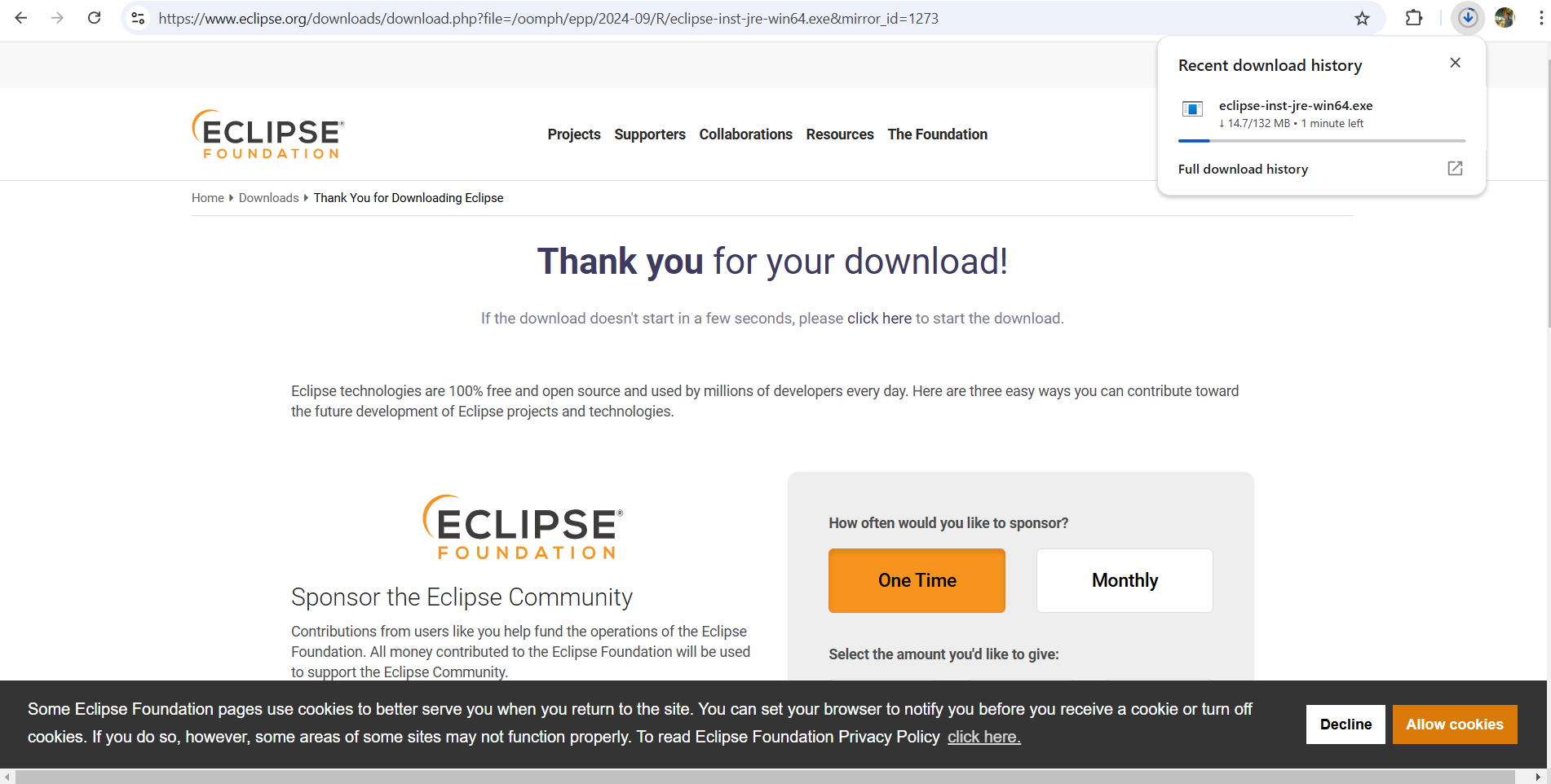
**Step 1: Download Eclipse Installer**:

->1.1 Go to the Eclipse Downloads page.



->1.2 Find "Eclipse IDE for Enterprise Java and Web Developers."

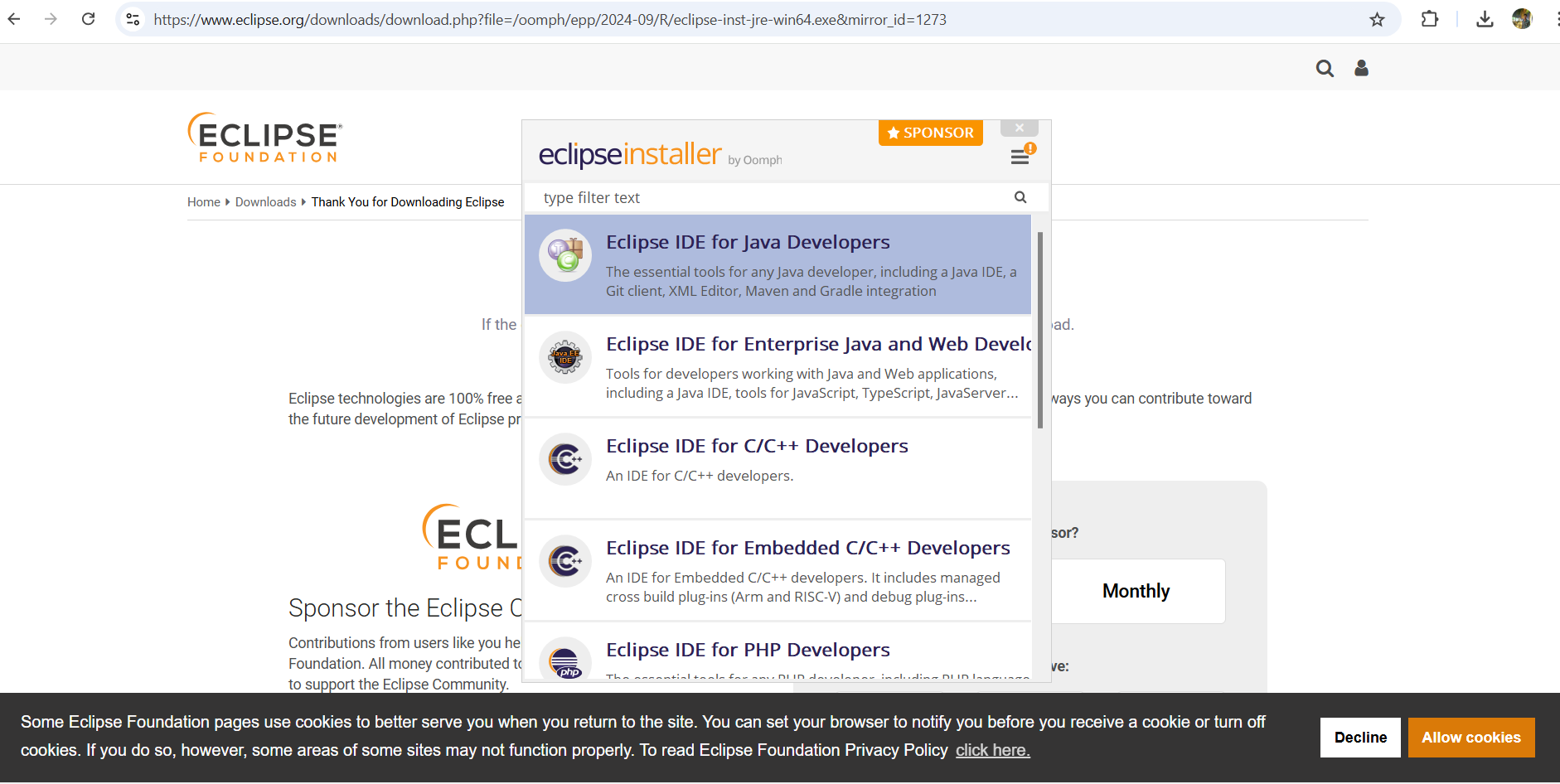
->1.3 Click on "Download x86\_64" to download the Windows installer.



**Step 2: Run the Eclipse Installer:**

-> 2.1 Locate the downloaded installer (eclipse-inst-jre-win64.exe).

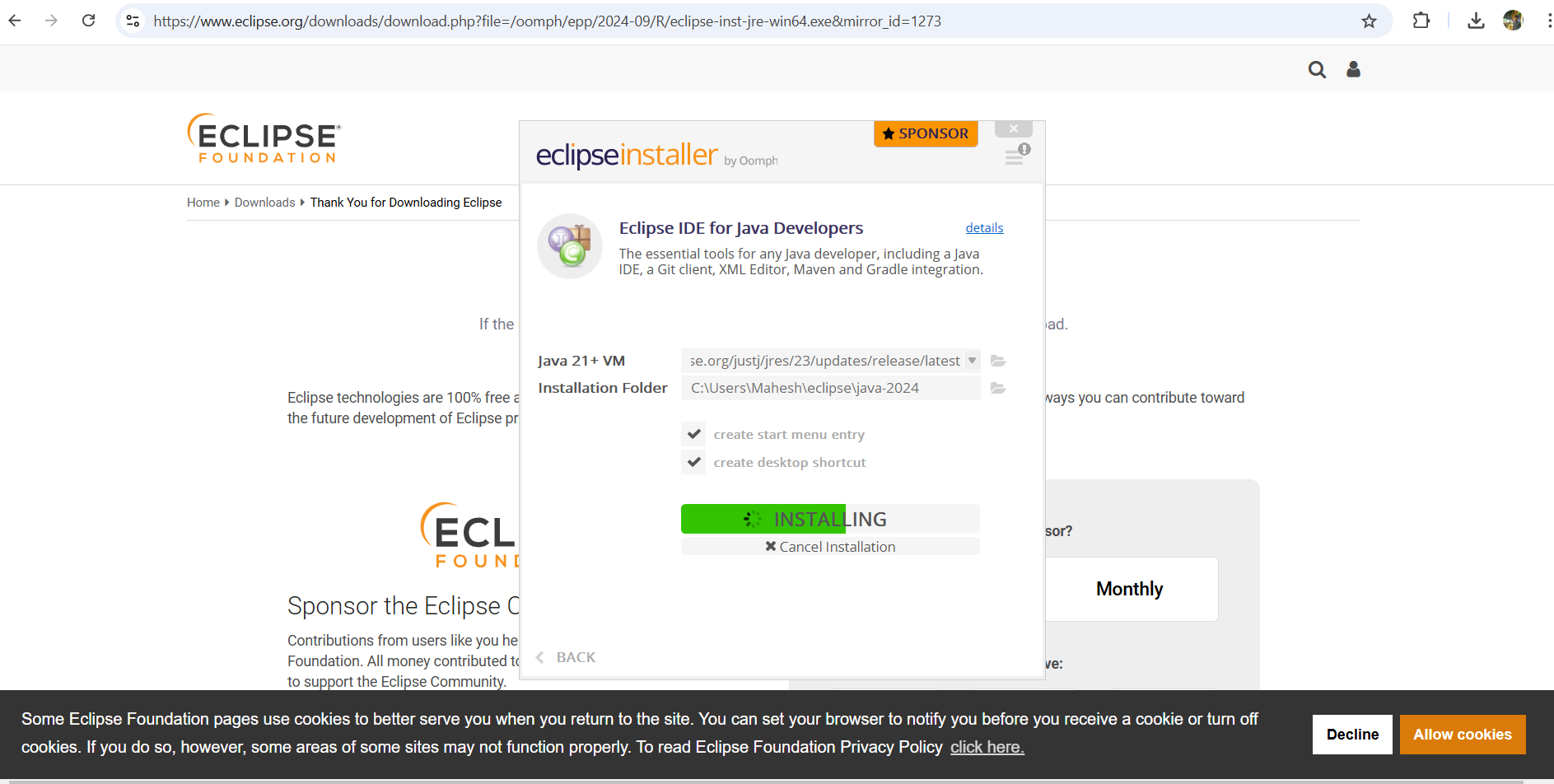
->2.2 Double-click the installer to run it.



->2.3 In the installer, choose:

-> "Eclipse IDE for Enterprise Java and Web Developers."

-> 2.4 Select the installation folder (use default or choose another).



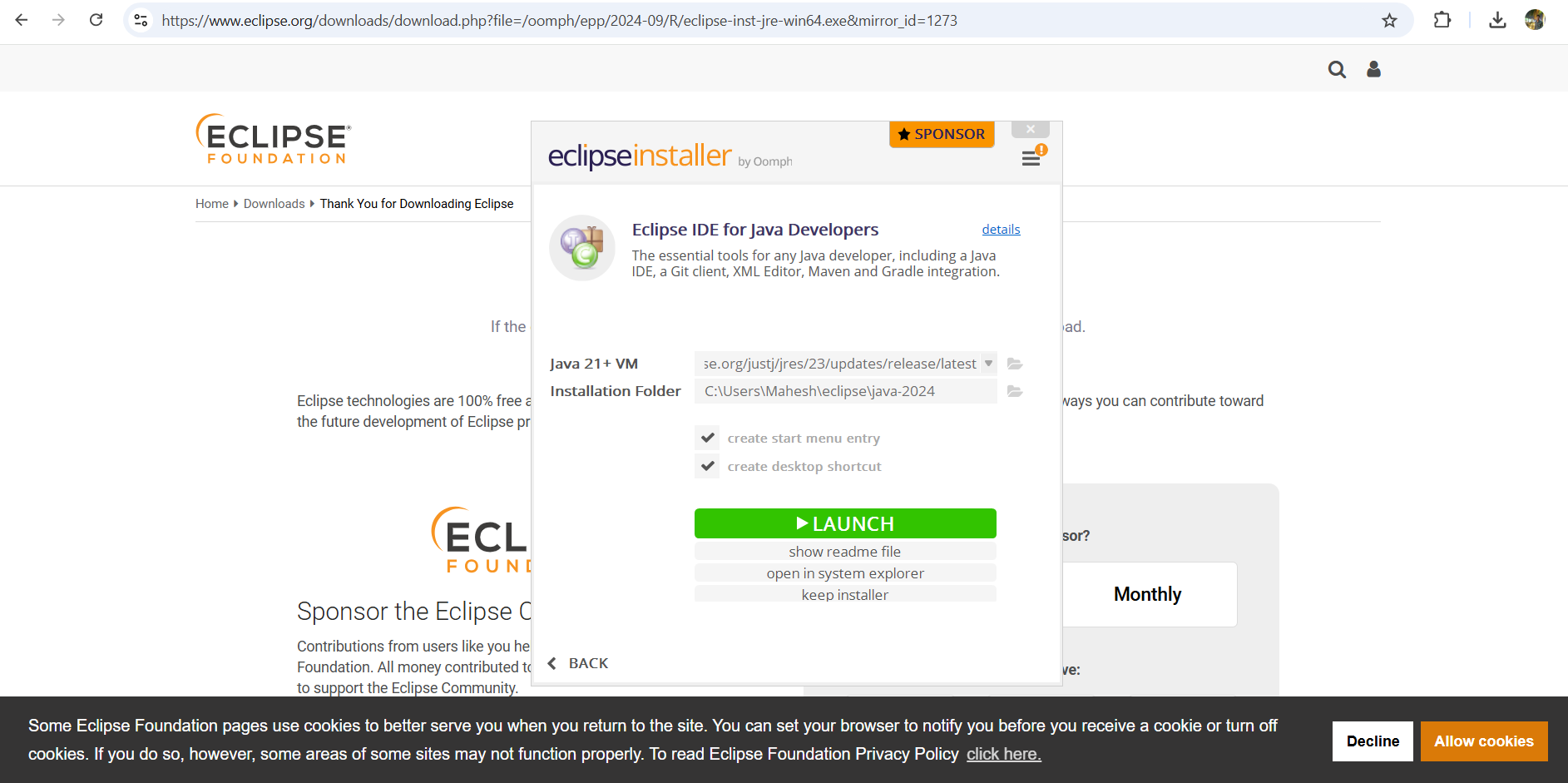
->2.5 Ensure the JDK path is correct (auto-detected or browse manually).

-> 2.6 Click "INSTALL" to begin the installation.

->2.7 Accept the license agreement to proceed.

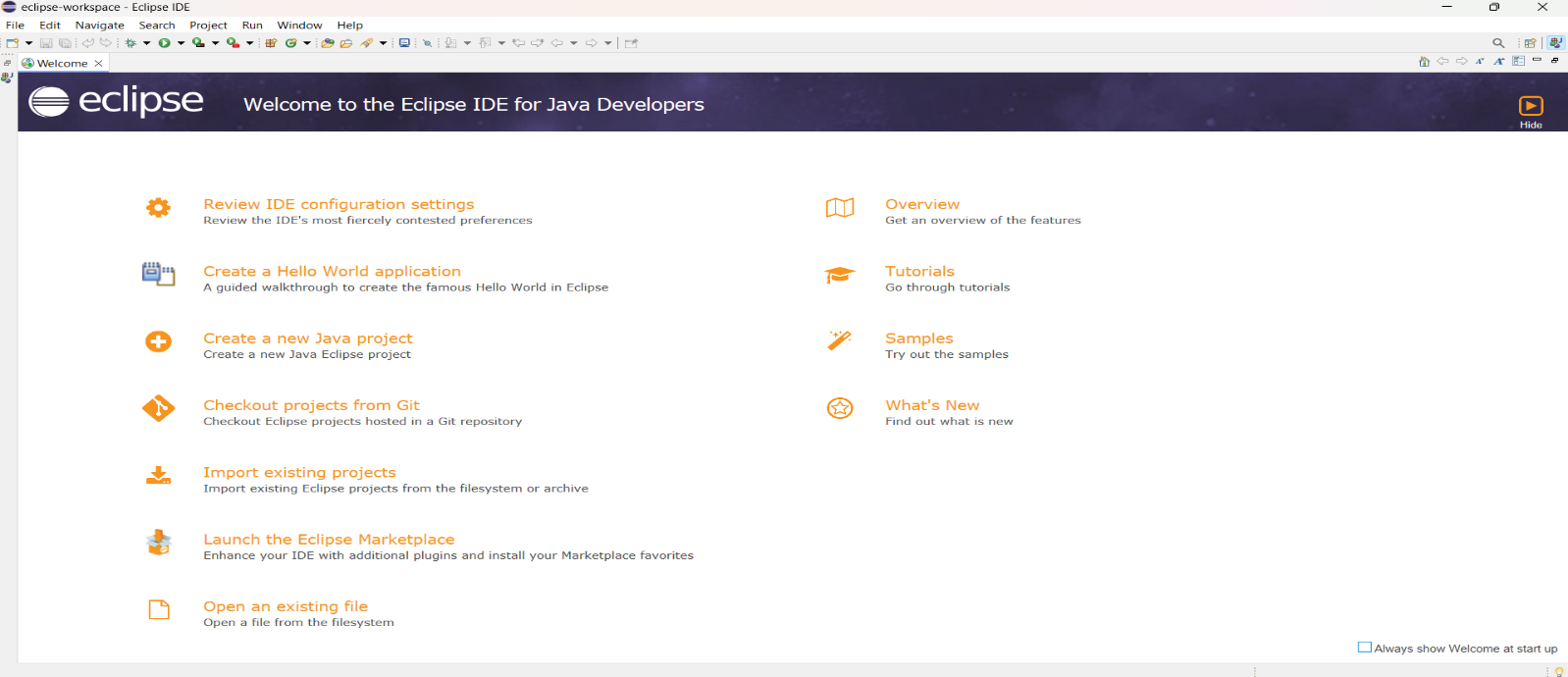
**Step 3: Launch Eclipse:**

->3.1 After installation, click "LAUNCH" to start Eclipse.



->3.2 Select a workspace folder for projects and settings (default or custom).

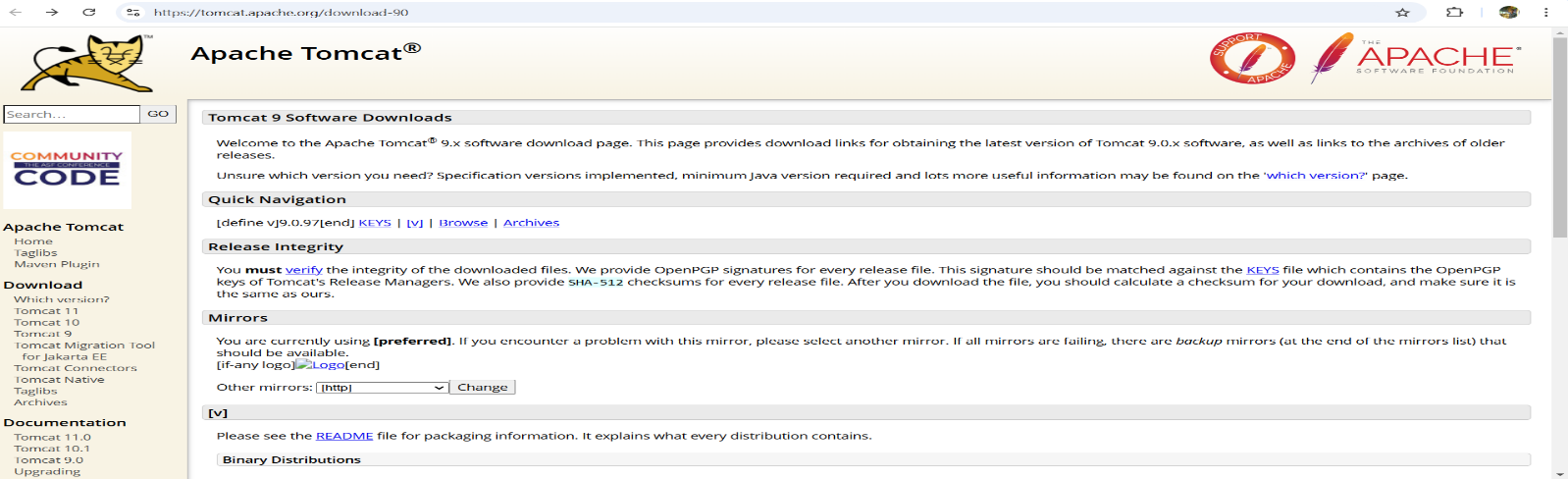
->3.3 Eclipse will start, and you're ready to create or import projects.



**2.2. Download Apache Tomcat 9.0**

**1. Download Apache Tomcat:**

-> 1.1 Go to Apache Tomcat 9.0 download page.

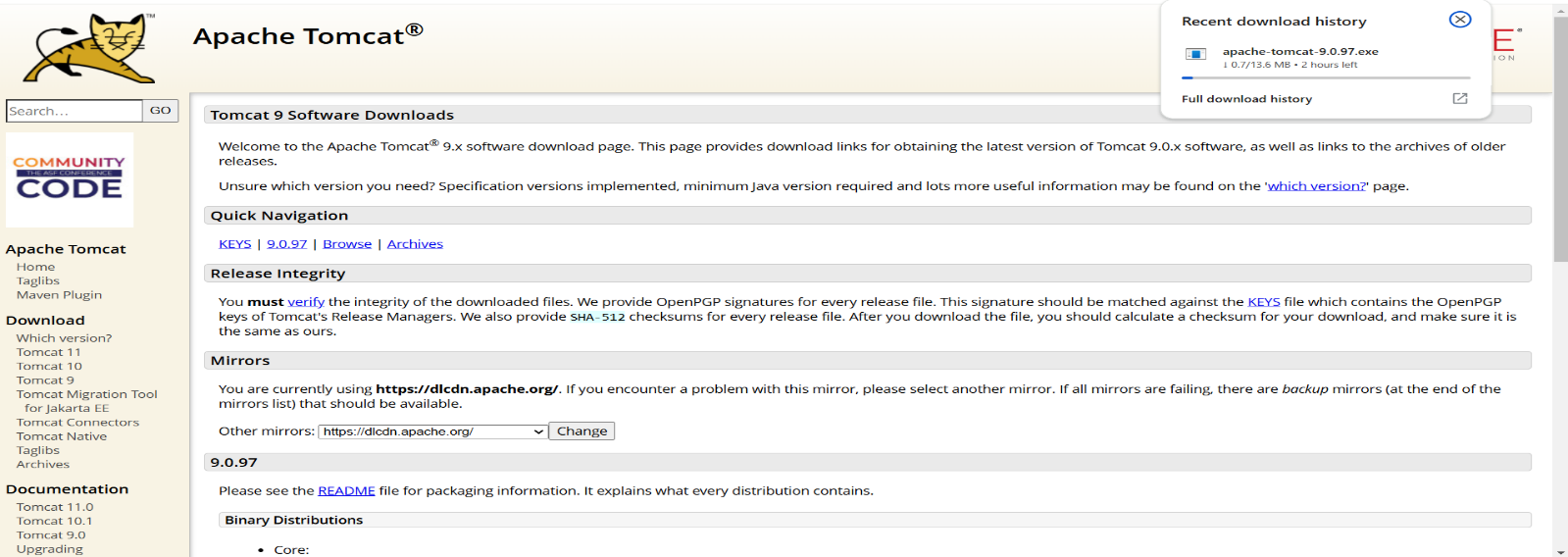


-> 1.2 Under "Binary Distributions" -> "Core" section, download the 32-bit/64-bit Windows Service Installer (.exe).

**2. Run the Installer:**

->2.1 Locate the .exe file and double-click to start the installation.

->2.2 Follow these installation steps:



->Welcome Screen -> Click Next.

-> License Agreement -> Accept and click Next.

-> Choose Components -> Leave defaults and click Next.

->Configuration:

-> Set the HTTP/1.1 Connector Port (default: 8080).

->Optionally, set Tomcat Administrator login.

->Click Next.

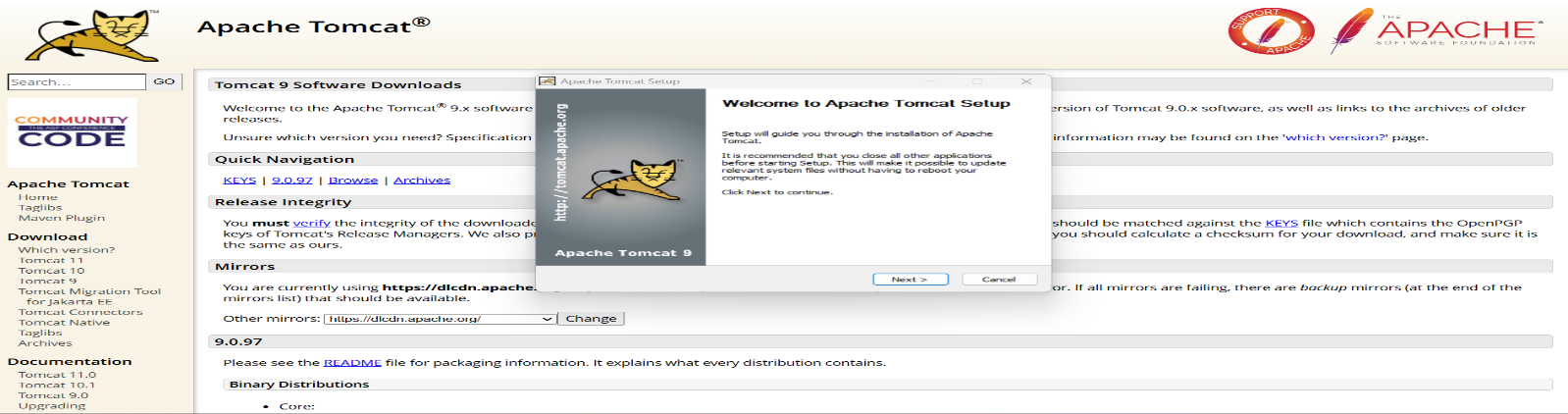
->Choose Java Virtual Machine -> Ensure correct JDK path.

->Install Location -> Choose location or leave default.

->Click Next, then Install.

**3. Complete Installation**:

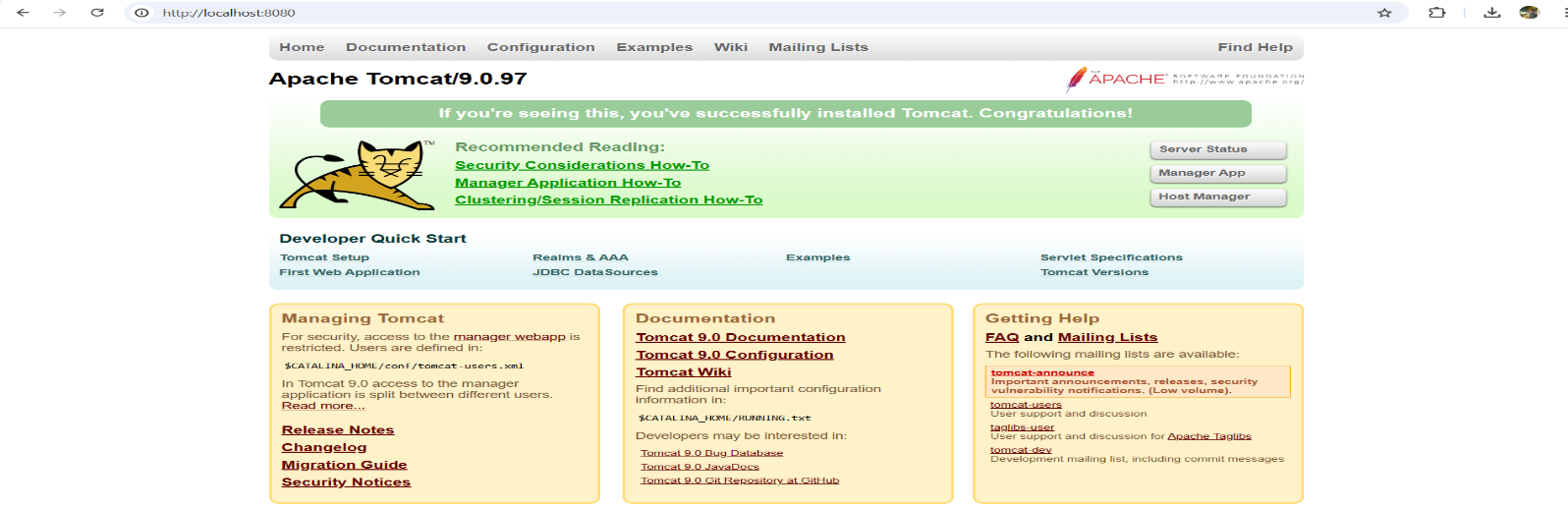
->3.1 After installation, choose to start Tomcat if desired.



-> 3.2 Click Finish to exit the installer.

**4. Verify Installation**:

->4.1 Open a web browser -> Go to <http://localhost:8080>.



->4.2 Tomcat homepage should display, confirming the installation.

**Tomcat Web-Server on Eclipse**

**1. Ensure You Have**:

-> Eclipse IDE for Java EE Developers installed.

-> Apache Tomcat 9.0 installed.

**2. Open Eclipse and Configure the Server**:

->2.1 Open Eclipse -> Go to Window -> Show View -> Other... -> Servers.

-> 2.2 Add a new server:

-> Right-click in Servers view -> New -> Server.

->Expand "Apache" -> Select "Tomcat v9.0" -> Click Next.

-> Set Server Name (optional).

->Browse to Tomcat installation directory (e.g., C:\Program Files\Apache Software Foundation\Tomcat 9.0).

->Ensure correct JRE is selected -> Click Finish.

**3. Add Projects to Server (optional):**

->3.1 In the wizard, add projects if desired or skip this step.

**4. Test Server Configuration:**

-> 4.1 Start the server:

->Right-click on the Tomcat server in the Servers view -> Select Start.

-> Console should show Tomcat is running.

->4.2 Access Tomcat:

-> Open a browser -> Go to http://localhost:8080.

-> Tomcat homepage confirms the server is running correctly.