# Glossary

## Introduction to Git and GitHub

### Terms and definitions of the course

## A

**Application Programming Interface (API) key:** This is an authentication token that calls an API, which is then called to identify the person, programmer, or program trying to access a website

## B

**Branch:** A pointer to a particular commit, representing an independent line of development in a project

## C

**CI/CD:** The name for the entire continuous integration and continuous deployment system

**Code reviews:** The deliberate and methodical gathering of other programmers to examine each other's code for errors to increase the code quality and reduces the amount of bugs

**Commit:** A command to make edits to multiple files and treat that collection of edits as a single change

**Commit files:** A stage where the changes made to files are safely stored in a snapshot in the Git directory

**Commit ID:** An identifier next to the word commit in the log

**Commit message:** A summary and description with contextual information on the parts of the code or configuration of the commit change

**Computer protocols:** Guidelines published as open standards so that any given protocol can be implemented in various products

**Continuous deployment (CD):** New code is deployed often after it has been automatically built and tested

**Continuous integration (CI):** A system that will automatically build and test our code every time there's a change

## D

**Diff:** A command to find the differences between two files

**Distributed:** Each developer has a copy of the whole repository on their local machine

**DNS zone file:** A configuration file that specifies the mappings between IP addresses and host names in your network

## F

**Fast-forward merge:** A merge when all the commits in the checked out branch are also in the branch that's being merged

**Fix up:** The decision to discard commit messages for that commit

**Forking:** A way of creating a copy of the given repository so that it belongs to our user

## G

**Git:** A free open source version control system available for installation on Unix based platforms, Windows and macOS

**GitHub:** A web-based Git repository hosting service, allowing users to share and access repositories on the web and copy or clone them to a local computer

**Git directory:** A database for a Git project that stores the changes and the change history

**Git log:** A log that displays commit messages

**Git staging area:** A file maintained by Git that contains all the information about what files and changes are going to go into the next commit

# H

**Head:** This points to the top of the branch that is being used

# I

**Indirect merges:** GitHub can merge a pull request automatically if the head branch is directly or indirectly merged into the base branch externally

**Issue tracker (bug tracker):** A tracker that shows tasks that need to be done, the state they're in and who's working on them

# M

**Master:** The default branch that Git creates for when a new repository initialized, commonly used to place the approved pieces of a project

**Merge:** An operation that merges the origin/master branch into a local master branch

**Merge commits:** All commits from the feature branch are added to the base branch

**Merge conflict:** This occurs when the changes are made on the same part of the same file, and Git won't know how to merge those changes

**Modified files:** A stage where changes have been made to a file, but the have not been stored or committed

# P

**Patch:** A command that can detect that there were changes made to the file and will do its best to apply the changes

**Pipelines:** The specific steps that need to run to obtain the desired result

**Pull request:** A procedure where new code is examined before it is merged to create a branch or master branch

**Private key:** A secret and secure cryptographic key that must be kept confidential and protected and is used to decrypt data that has been encrypted with the corresponding public key

**Public key**: A safety cryptographic structure frequently employed to establish secure communication through data encryption or to validate the authenticity of a digital signature

# R

**Rebasing:** The base commit that's used for a branch is changed

**Remote branches:** Git uses read-only branches to keep copies of the data that's stored in the remote repository

**Remote repositories:** Repositories that allow developers to contribute to a project from their own workstations making changes to local copies of the project independently of one another

**Repository:** An organization system of files that contain separate software projects

**Rollback:** The act of reverting changes made to software to a previous state

# S

**Source Control Management (SCM):** A tool similar to VCS to store source code

**Stage files:** A stage where the changes to files are ready to be committed

**Secure Shell (SSH):** A robust protocol for connecting to servers remotely

**SSH client:** This  establishes a connection to the SSH server, ensuring a secure interaction, where the client makes access requests

**SSH key:** An access credential

**SSH protocol:** Standard commonly used for logging in to servers remotely on the principle of public-key encryption

**SSH server:** This establishes secure network connections, undergoes mutual authentication, and initiates encrypted login sessions or file transfers

**Squash commits:** The decision add commit messages together and an editor opens to make any necessary changes

# T

**Three-way merge:** A merge when the snapshots at the two branch tips with the most recent common ancestor, the commit before the divergence

**Tracked:** A file's changes are recorded

## U

**Untracked:** A file's changes are not recorded

## V

**Version control systems (VCS):** A tool to safely test code before releasing it, allow multiple people collaborate on the same coding projects together, and stores the history of that code and configuration