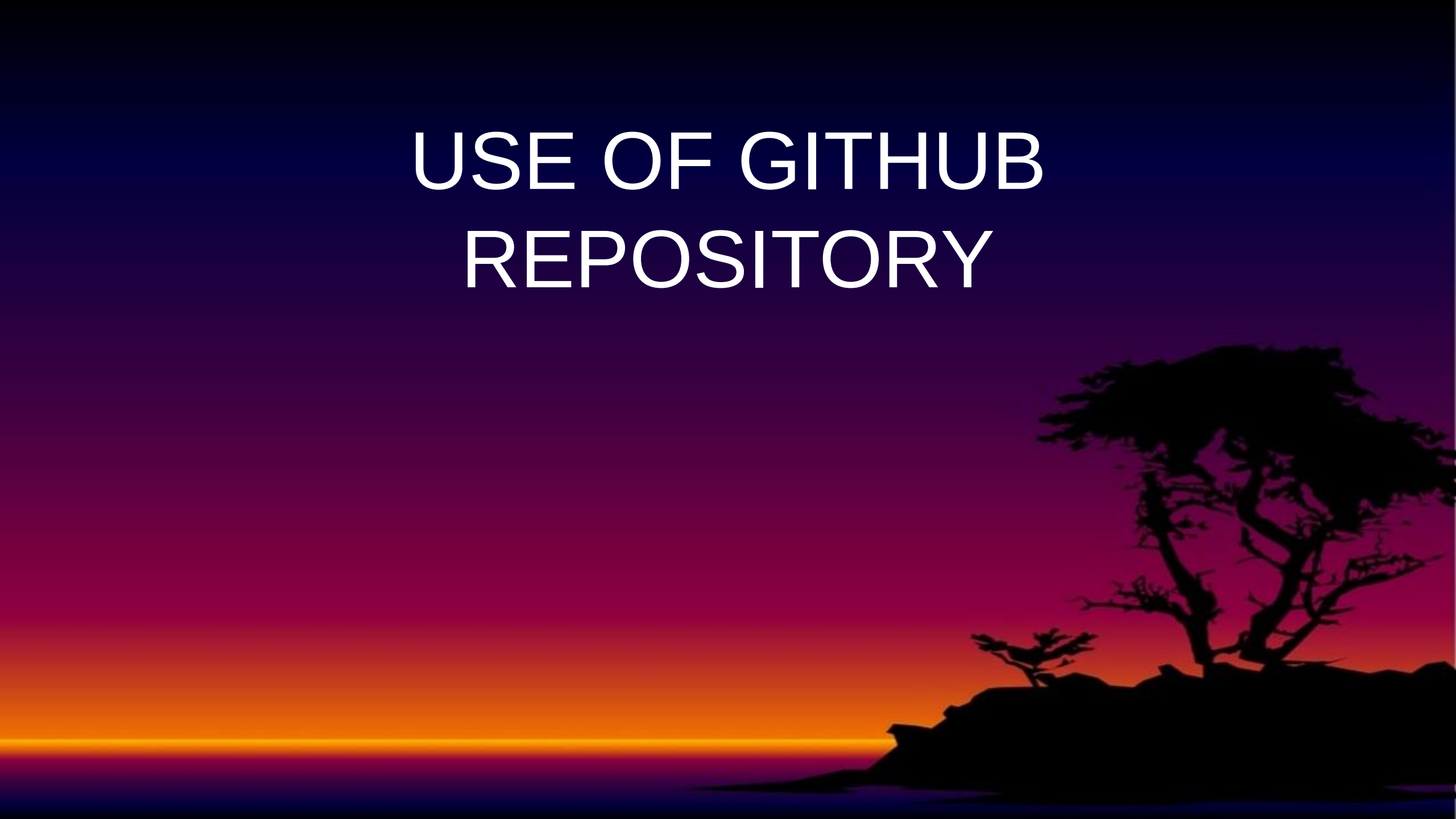
USE OF GITHUB REPOSITORY



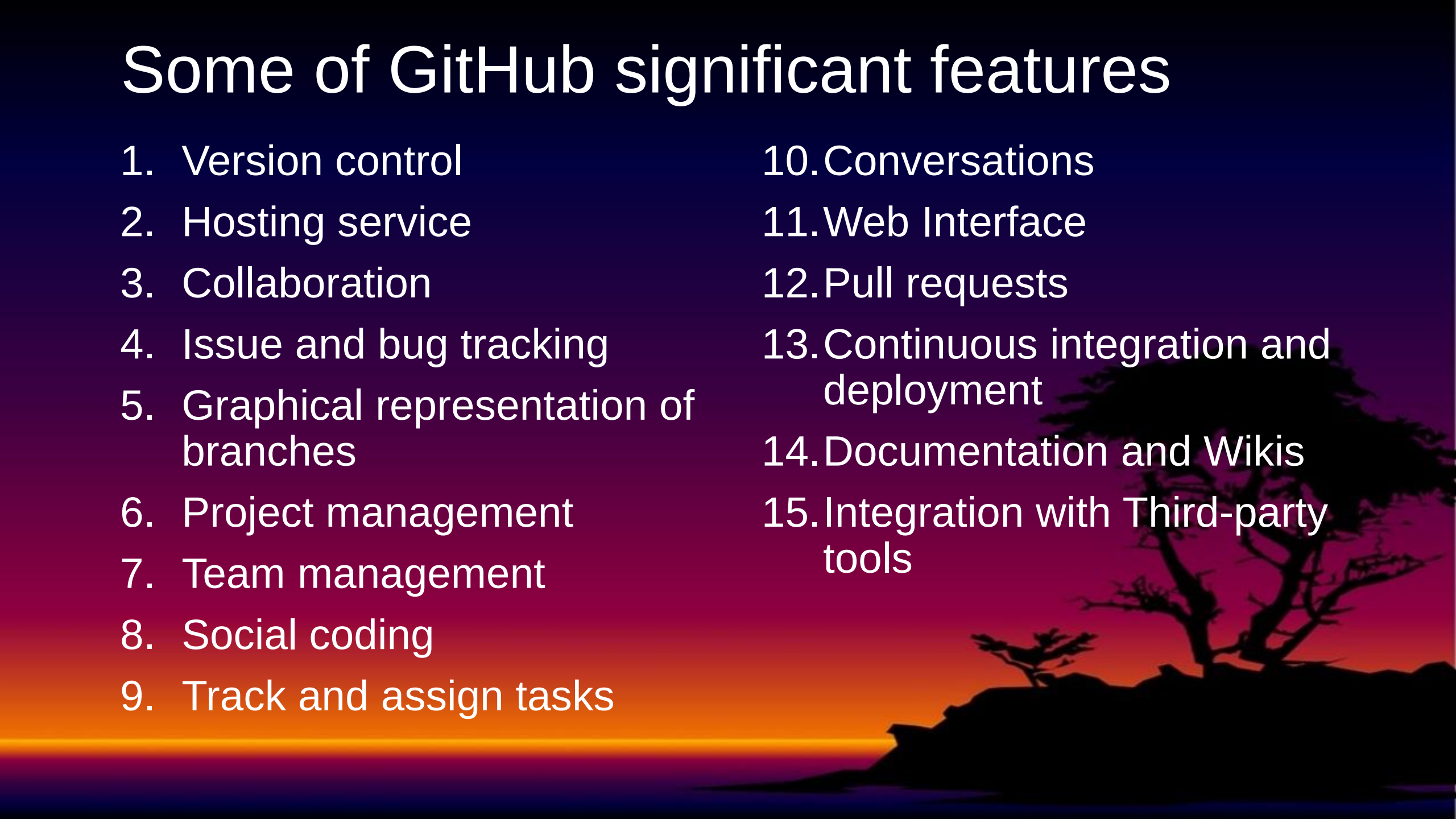
USE OF GITHUB REPOSITORY

Description of GitHub

- Github is an Internet hosting service for software development and version control using Git.
- GitHub is a Git repository hosting service. GitHub also facilitates with many of its features, such as access control and collaboration.
- It provides a Web-based graphical interface.
- GitHub is an American company. It hosts source code of your project in the form of different programming languages and keeps track of the various changes made by programmers.

Some of GitHub significant features

1. Version control
2. Hosting service
3. Collaboration
4. Issue and bug tracking
5. Graphical representation of branches
6. Project management
7. Team management
8. Social coding
9. Track and assign tasks
10. Conversations
11. Web Interface
12. Pull requests
13. Continuous integration and deployment
14. Documentation and Wikis
15. Integration with Third-party tools



The key benefits of GitHub

- It is easy to contribute to open source projects via GitHub.
- It helps to create an excellent document.
- You can attract recruiter by showing off your work. If you have a profile on GitHub,
- you will have a higher chance of being recruited.
- It allows your work to get out there in front of the public.
- You can track changes in your code across versions.

Create account on GitHub

1. Open https://github.com in a web browser, and then select Sign up.
2. Enter your email address.
3. set a password and username
4. Verify your account by solving a puzzle.
5. select the Create account button
6. Type launch code(from verification email) in the Enter code dialog, and then press Enter

Create new remote repository

Step 1:

- Sign in to your GitHub account: Go to the GitHub homepage (github.com) and sign in with your username and password.

Step 2:

- Access the repository creation page: Once you're signed in, click on the "+" icon in the top-right corner of the GitHub page. From the dropdown menu, select "New repository." Alternatively, you can directly visit the repository creation page by going to github.com/new.

Step 3:

- Fill required information such as Repository name and Description and click on create repository.

Step 4:

• Create the repository: Once you've provided the required information and selected any desired options, click on the "Create repository" button. GitHub will now create your new remote repository.

Step 5:

• Repository setup: After creating the repository, you'll be redirected to the repository page. Here, you can further configure and manage your repository, such as adding files, creating branches, setting up project boards, and inviting collaborators.

Apply git commands related to repository
Git clone

- The **"git clone"** command is used to create a local copy of a

remote repository. It allows you to download all the files and commit

history from a remote Git repository to your local machine.

- Typically, the "original" repository is located on a remote server, often from a service like GitHub, Bitbucket, or GitLab).
- That remote repository's URL is then later referred to as the "origin".

Here is how you can use the "git clone" command:

Step 1: Open your terminal(git bash) or command prompt: Launch the terminal or

command prompt application on your local machine.

Step 2: Navigate to the directory where you want to clone the repository:

- Use the "cd" command to change to the desired directory. For
- example, if you want to clone the repository to your "Documents"

folder, you can use the following command:

cd Documents

Step 3: Clone the repository: To clone a remote repository, you need the URL of the repository. The URL can be found on the repository's GitHub page. Use the following command to clone the repository:

git clone <repository-url>

Replace "<repository-url>" with the URL of the repository you want to clone. For example:

git clone https://github.com/username/repository-name.git

The repository will be cloned into a new directory with the same name as the remote repository.

• If you want to specify a different directory name, you can add it as an additional argument at the end of the command.

 Authenticate (if required): If the repository is private and requires authentication, Git will prompt you to provide your GitHub username and password or a personal access token. Enter the credentials to proceed with the cloning process.

 Wait for the cloning process to complete

 Repository cloned successfully: Once the cloning process is complete, you will see a message indicating that the repository has been cloned successfully

Usage Examples

- In its simplest (and most common) form, only the repository URL is specified:
- cd folder/to/clone-into/
- git clone https://github.com/gittower/git-crash-course.git
- This will download the project to a folder named after the Git repository ("git-crash-course" in this case). If you want a different folder name, simply specify it as the last parameter:
- git clone https://github.com/gittower/git-crash-course.git other-name

Git remote

- The "git remote" command in Git is used to manage the remote repositories associated with your local repository. It allows you to view, add, rename, or remove remote repositories
- View remote repositories:

git remote

- This command will list the names of all remote repositories associated with your local repository. By default, the primary remote repository is named "origin."

- Show detailed information about a specific remote repository:

git remote show <remote-name>

Replace "<remote-name>" with the name of the remote repository. This command displays information such as the remote URL, the branches tracked by the remote repository, and more.

- Add a remote repository:

git remote add <remote-name> <repository-url>

Replace "<remote-name>" with a name of your choice to identify the remote repository and "<repository-url>" with the URL of the remote repository you want to add. This allows you to connect your local repository with the remote repository.

- Rename a remote repository:

git remote rename <old-name> <new-name>

Replace "<old-name>" with the current name of the remote repository and "<new-name>" with the new name you want to assign to it.

- Remove a remote repository:

git remote remove <remote-name>

Replace "<remote-name>" with the name of the remote repository you want to remove. This command disconnects your local repository from the remote repository.

- Update the URL of a remote repository:

git remote set-url <remote-name> <new-url>

Replace "<remote-name>" with the name of the remote repository and "<new-url>" with the new URL you want to set for the remote repository. This can be useful if the remote repository's URL has changed.