

Agenda :

- Git Command

✓- Staging Area - add, status , commit , log , push , remote.

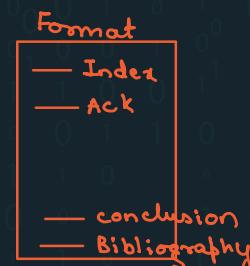
✓- Branches - Creation, switch , merge , deleting

✓- Git merge Conflict. -> How we can resolve that conflict.

✓- Github Basic Exercise

✓- Readme.Md [Markdown Language]. {Describe how to use the project that was upload on the Github} [Documentation]

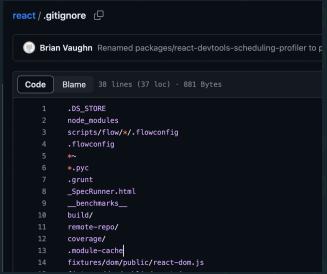
-ChatGPT [Helps] -> Format



-Figma [Classroom] : UI/UX

-o HackerRank Assignment : Java DSA

| Command | Description |
|----------------------|--|
| pwd | Present Working Directory |
| mkdir | Make a directory or a folder |
| cd | Move into particular directory |
| cd .. | Move back to the previous directory |
| ls | list all the directories/ sub directories / file |
| ls -la | list down all the hidden files |
| touch <file_name> | To create the file |
| cat <file_name> | To read file data |
| nano <file_name> | To open the file / write. [modified] |
| Copy Command <cp> | To copy the file from one path to another path |

| | |
|--|--|
| <p>mv - Rename</p> <p>mv - Move</p> <p>rmdir <dir_name> rm <file_name> rm -r <file_name></p> | <p>To rename the file <old_name> <new></p> <p>Move the file from one directory to another by providing path <loc1> to <loc2></p> <p>To remove the empty directory To remove the file To remove the content forcefully.</p> |
| <p>Git Commands</p> <p>git --version</p> <p>git config --global user.name "name", user.email "email";</p> <p>git init</p> <p>git status</p> <p>git add <file> <file></p> <p>git add .</p> <p>.gitignore don't want to track .</p> | <p>It reflects the current version of git available in my system. Authentication -> configure the git.</p> <p>Initialise the file with .git -> to track all the changes that are being done on that particular repo.</p> <p>Display the current status of your modified file.</p> <p>Will add the selected file to the staging area. You Know how to un-stage a file 😬 😬. Adds all the file that are available in my untracked status.</p> <p>I put all the files[secret / unwanted] file that we don't want to track .</p>  |

| | |
|---|--|
| <code>git commit -m</code> | It will add the file to Local Repo [Check point] |
| <code>git log</code> | Shows all the history of commits. |
| <code>git remote -v</code> | To view the available remote file |
| <code>git remote add origin <url></code> | To link the remote repo with my local repo |
| <code>git push -u origin main</code> | To push file from local to remote |
| <code>git push origin main</code> | Now upstream is already. You can run this command from 2nd time onwards. |
| <code>git branch / git branch --list</code> | To check the available branch |
| <code>git branch <name></code> | Create a branch |
| <code>git switch <name></code> | Switch to another branch |
| <code>git switch -c <name></code> | Create and switch at the same time |
| <code>git checkout <name></code> | To Switch between the branches. |
| <code>git checkout -b <name></code> | To Create and switch to the other branch at the same time. |
| <code>git branch -d <name></code> | To delete the branch |
| <code>git branch -D <name></code> | To delete it forcefully |
| <code>git merge <branch_name></code> | First switch to other branch where you need to merge the specific branch that you provided in the command. |
| <code>git diff</code> | To check the difference in the modified file |
| <code>git clone <url></code> | To add Github repo to our local repo. |

Github Basic Exercise.

<https://geekster-in.notion.site/Github-Basics-Exercise-75046819394246c882437f9b06c2f140>

Github Basics Exercise

1. Create a new repository locally on your machine.
2. Create a new Github repository. Name it whatever you want.
3. Connect your local repo to the Github repo.
4. **Optional:** rename the default branch from master to main.
5. Make a new file called `favorites.txt`. Leave it empty. Make your first commit on the `main` branch.
6. Push up your `main` branch to Github! Make sure you see your empty `favorites.txt` file on Github.
7. Next, create two branches: `foods` and `movies`.
8. Switch to the `foods` branch. Add three (or more) of your favorite foods to the `favorites.txt` file. Add and commit your changes on the `foods` branch.
9. Switch to the `movies` branch and add three or more of your favorite movies to the `favorites.txt` file. Add and commit your changes on the `movies` branch.
10. Push up your `foods` branch to Github. Make sure you see it on Github!
11. Push up your `movies` branch to Github. Make sure you see it on Github!
12. Merge the `foods` branch into the `main` branch. Then merge the `movies` branch into the `main` branch. If necessary, resolve conflicts so that you end up with your favorite foods and favorite movies in the same `favorites.txt` file.
13. Push up the latest work on your `main` branch to Github.

```
hub > favorites.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
1 <<<<< HEAD (Current Change)
2 Biryani
3 Pizza
4 Butter Chicken
5 Momos
5 Pav Bhaji
7 Dal Makhani
3 Panner Tika
3 Butter Nan
3 Kabab
L Chole Bhature
2 Allu Paratha
3 =====
4 3 Idiots
5 Tere Naam
5 RRR
7 PK
3 Kargil
3 KGF
3 Chichore
L Bahubali
2 Tiger 3
3 Animal
4 Saalar
5 >>>> movies (Incoming Change)
```

Solution :

```

@Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise (main) $ mkdir Github
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise (main) $ cd Github/
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git status
  On branch main
    Your branch is up to date with 'origin/main'.

      nothing to commit, working tree clean
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git init
  Initialized empty Git repository in /workspaces/Committing_Basic_Exercise/Github/.git/
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git remote add origin https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git remote -v
  origin https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git (fetch)
  origin https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git (push)
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git branch
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git branch --list
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ touch favorites.txt
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git status
  On branch main
    No commits yet

  Untracked files:
    (use "git add <file>..." to include in what will be committed)
      favorites.txt

  nothing added to commit but untracked files present (use "git add" to track)
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git add favorites.txt
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git commit -m "committing empty favorites.txt"
[main (root-commit) 3b7e3e9] committing empty favorites.txt
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 favorites.txt
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git log
commit 3b7e3e96ea4855ca502b93d36e1a8bc7ca42bf (HEAD -> main)
Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
Date:   Fri Dec 22 15:36:36 2023 +0000

  committing empty favorites.txt

@Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git push -u origin main
To https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git
  ! [rejected]          main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git push --force origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 248 bytes | 248.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git
  + b0bcc7...3b7e3e9 main -> main (forced update)
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git branch
* main
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git branch foods
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git branch movies
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git branch
  foods
* main
  movies
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (main) $ git switch foods
Switched to branch 'foods'
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (foods) $ git branch
* foods
  main
  movies
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (foods) $ git status
On branch foods
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   favorites.txt

no changes added to commit (use "git add" and/or "git commit -a")
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (foods) $ git add favorites.txt
● @Krishna-Geekster ~ /workspaces/Committing_Basic_Exercise/Github (foods) $ git commit -m "Adding 11 best quality foods"

```

```

● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git add favorites.txt
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git commit -m "Adding 11 best quality foods"
[foods 1956c78] Adding 11 best quality foods
  1 file changed, 11 insertions(+)
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git log
commit 1956c78d9a0a7ff60df68f890fe200f7e260b7bb (HEAD -> foods)
Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
Date:   Fri Dec 22 15:41:09 2023 +0000

    Adding 11 best quality foods

    commit 3b7e3e964ea4855ca502b93d36e1a8bc7ca4a2bf (origin/main, movies, main)
    Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
    Date:   Fri Dec 22 15:36:36 2023 +0000

        committing empty favorites.txt
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git switch main
Switched to branch 'main'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git checkout foods
Switched to branch 'foods'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git switch movies
Switched to branch 'movies'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git branch
  foods
  main
* movies
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git status
On branch movies
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   favorites.txt

no changes added to commit (use "git add" and/or "git commit -a")
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git add favorites.txt
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git status
On branch movies
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   favorites.txt

● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git commit -m "Adding my 11 favorites movies"
[movies fe84c8b] Adding my 11 favorites movies
  1 file changed, 11 insertions(+)
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git log
commit fe84c8b2f000919da7f8bf3bc507c43e4844c08 (HEAD -> movies)
Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
Date:   Fri Dec 22 15:43:37 2023 +0000

    Adding my 11 favorites movies

    commit 3b7e3e964ea4855ca502b93d36e1a8bc7ca4a2bf (origin/main, main)
    Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
    Date:   Fri Dec 22 15:36:36 2023 +0000

        committing empty favorites.txt
@Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git main
git: 'main' is not a git command. See 'git --help'.

The most similar command is
  mailinfo
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git switch main
Switched to branch 'main'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git switch foods
Switched to branch 'foods'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git switch movies
Switched to branch 'movies'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git switch foods
Switched to branch 'foods'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git push origin foods
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 374 bytes | 374.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'foods' on GitHub by visiting:
remote:     https://github.com/Krishna-Geekster/Committing_Basic_Exercise/pull/new/foods
remote:
To https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git

```

```

● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git branch
  * foods
    main
    movies
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (foods) $ git switch movies
Switched to branch 'movies'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git push origin movies
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 356 bytes | 356.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
  Create a pull request for 'movies' on GitHub by visiting:
  remote:     https://github.com/Krishna-Geekster/Committing_Basic_Exercise/pull/new/movies
remote:
  To https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git
    * [new branch]      movies -> movies
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (movies) $ git switch main
Switched to branch 'main'
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git merge foods
Updating 3b7e3e9..1956c78
Fast-forward
  favorites.txt | 11 ++++++++-
  1 file changed, 11 insertions(+)
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git branch
  foods
  * main
  movies
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git merge movies
Auto-merging favorites.txt
CONFLICT (content): Merge conflict in favorites.txt
Automatic merge failed; fix conflicts and then commit the result.
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git stauts
git: 'stauts' is not a git command. See 'git --help'.

The most similar command is
  status

```

```

● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git status
On branch main
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:  favorites.txt

no changes added to commit (use "git add" and/or "git commit -a")
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git add favorites.txt
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git commit -m "Resolved Conflicts and happy with both the data"
[main c77a41e] Resolved Conflicts and happy with both the data
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git log
commit c77a41ee52566692576ffdf87811a83c3668e911 (HEAD -> main)
Merge: 1956c78 fe84c8b
Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
Date:   Fri Dec 22 15:52:13 2023 +0000

        Resolved Conflicts and happy with both the data

commit fe84c8b2f7000919da7f8bf3bc507c43e4844c08 (origin/movies, movies)
Author: Krishna <122261087+Krishna-Geekster@users.noreply.github.com>
Date:   Fri Dec 22 15:43:37 2023 +0000

● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 469 bytes | 469.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Krishna-Geekster/Committing_Basic_Exercise.git
  3b7e3e9..c77a41e main -> main
● @Krishna-Geekster → /workspaces/Committing_Basic_Exercise/Github (main) $ 

```

Ln 3, Col 15 Spaces: 4 UTF-8 LF Plain Text Layout: U.S. □

Readme.Md [MarkDown Language]

| Markdown | HTML |
|----------------------------------|---|
| # Heading 1 | <h1>Heading 1</h1> |
| ## Heading 2 | <h2>Heading 2</h2> |
| **bold text** | bold text |
| *italic text* | <i>italic text</i> |
| - Item 1 | Item 1 |
| 1. Item 1 | Item 1 |
| [Google](https://www.google.com) | Google |
| ![[image]](https://image.url) | |
| ```code``` | <pre>code</pre> |

```
import { createRoot } from 'react-dom/client';

function HelloMessage({ name }) {
  return <div>Hello {name}</div>;
}

const root = createRoot(document.getElementById('container'));
root.render(<HelloMessage name="Taylor" />);
```

favorites.txt | README.MD | preview README.MD

① README.MD > # Indian Premier League > ## Examples

```
1 # Indian Premier League
2
3 ## React
4 **React** is a JavaScript library for building user interfaces.
5
6 **Declarative:** React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.
7
8 **Component-Based:** Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.
9
10 **Learn Once, Write Anywhere:** We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node and power mobile apps using [React Native](https://reactnative.dev/).
11
12 ## Installation
13
14 React has been designed for gradual adoption from the start, and you can use as little or as much React as you need:
15
16 - Use [Quick Start](https://react.dev/learn) to get a taste of React.
17 - [Add React to an Existing Project](https://react.dev/learn/add-react-to-an-existing-project) to use as little or as much React as you need.
18 - [Create a New React App](https://react.dev/learn/start-a-new-react-project) if you're looking for a powerful JavaScript toolchain.
19
20 ## Documentation
21 You can find the React documentation on the website.
```

② Indian Premier League

React

React is a JavaScript library for building user interfaces.

Declarative: React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable, simpler to understand, and easier to debug.

Component-Based: Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.

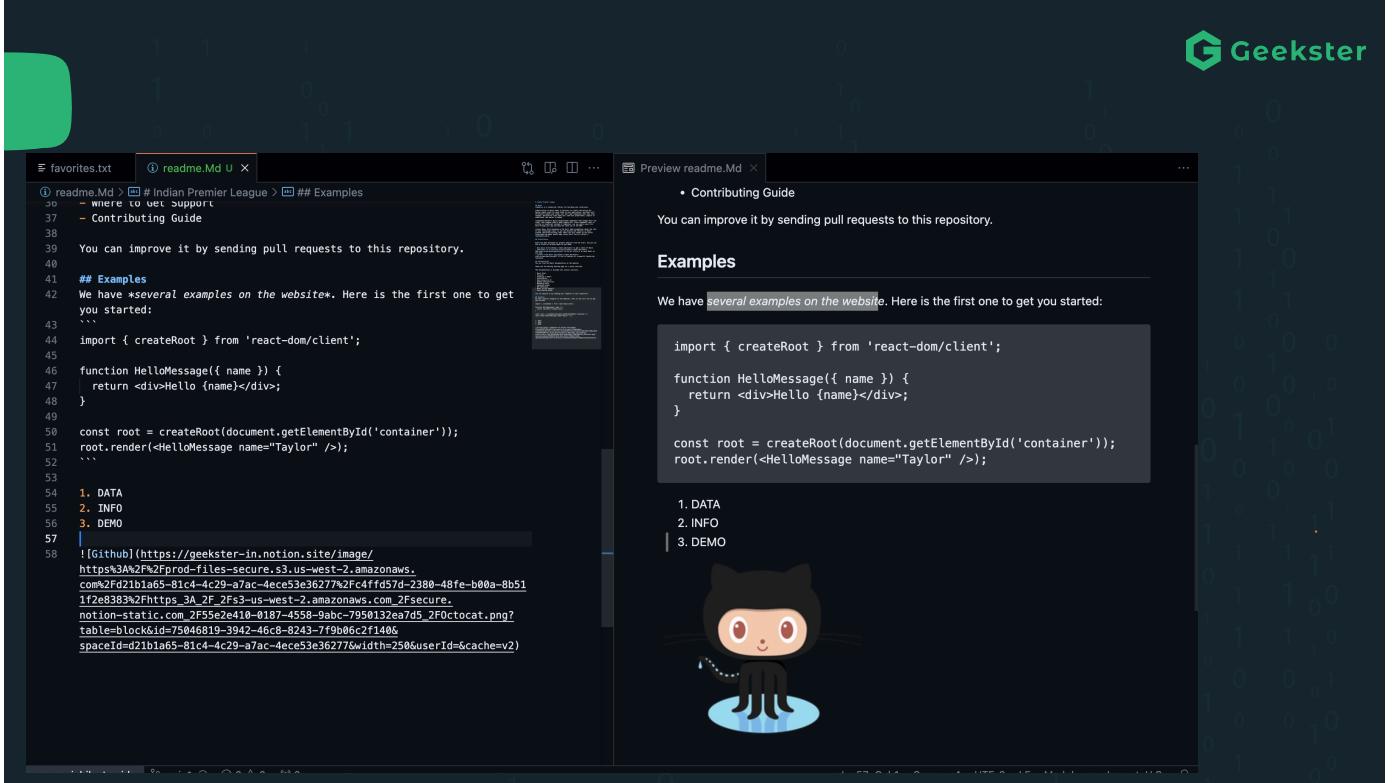
Learn Once, Write Anywhere: We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node and power mobile apps using **React Native**.

Installation

React has been designed for gradual adoption from the start, and you can use as little or as much React as you need:

- Use [Quick Start](#) to get a taste of React.
- Add React to an Existing Project to use as little or as much React as you need.
- Create a New React App if you're looking for a powerful JavaScript toolchain.

Ln 57, Col 1 Spaces: 4 UTF-8 LF Markdown Layout: U.S. Q



Chat GPT : Format of Readme File.