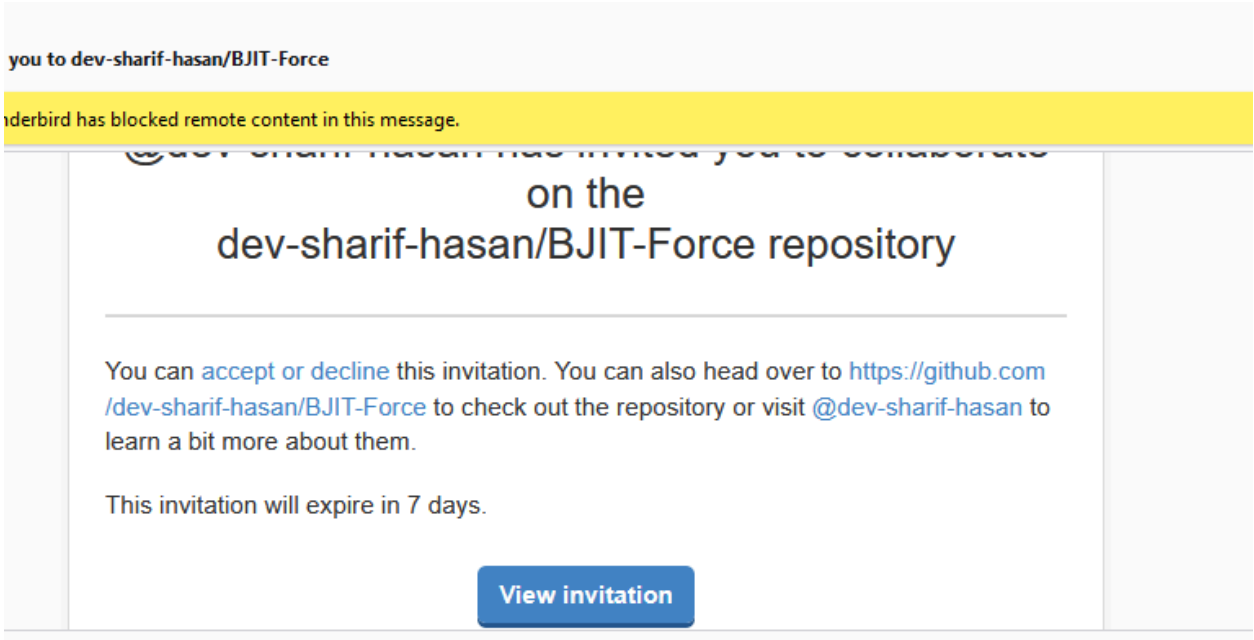
1. Accepting the invitation link



2. Cloning the repo using SSH key

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c
$ git clone git@github.com:dev-sharif-hasan/BJIT-Force.git
Cloning into 'BJIT-Force'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

3. Navigate the folder

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c
$ cd BJIT-Force
```

4. Pull the project

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git pull origin development
From github.com:dev-sharif-hasan/BJIT-Force
 * branch            development -> FETCH_HEAD
Already up to date.
```

5. Creating branch and navigate the task3branch

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git branch task3branch

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git branch
* development
  task3branch

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git checkout task3branch
Switched to branch 'task3branch'
```

6. Checking current status

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3branch)
$ git status
On branch task3branch
nothing to commit, working tree clean
```

7. After modifying the README.md file again check the status

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3branch)
$ git status
On branch task3branch
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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

8. Add and commit with meaningful message

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3branch)
$ git add README.md

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3branch)
$ git status
On branch task3branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```

9. Commit the README.md file

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3branch)
$ git commit --amend -m "Doc(README.md): Adding prerequisite and installation" -m "Before starting it's important to knoww
hat prerequisite is needed and intallation processs" -m "Task #3"
[task3branch 9cf17af] Doc(README.md): Adding prerequisite and installation
 Date: Mon Nov 25 14:07:27 2024 +0600
 1 file changed, 23 insertions(+), 2 deletions(-)
```

10. Checking the git log history

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3branch)
$ git log
commit 9cf17af6cba8be2133bfccc1468a1080cacdf059 (HEAD -> task3branch)
Author: Nasrul Hasan <nasrul.hasan@bjitacademy.com>
Date:   Mon Nov 25 14:07:27 2024 +0600

    Doc(README.md): Adding prerequisite and installation

    Before starting it's important to knowwhat prerequisite is needed and intallation processs

    Task #3

commit 905a5d97e72e76e90a36adeb9e91a2c7c284671b (origin/main, origin/development, origin/HEAD, development)
Author: dev-sharif-hasan <sharif.hasan@bjitacademy.com>
Date:   Mon Nov 25 12:14:54 2024 +0600

    Initial commit
```

11. Create new branch Task3

```
 README.md | 54 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++--
 1 file changed, 52 insertions(+), 2 deletions(-)

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git chekout task3
git: 'chekout' is not a git command. See 'git --help'.

The most similar command is
        checkout

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git checkout -b task3
Switched to a new branch 'task3'

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3)
$
```

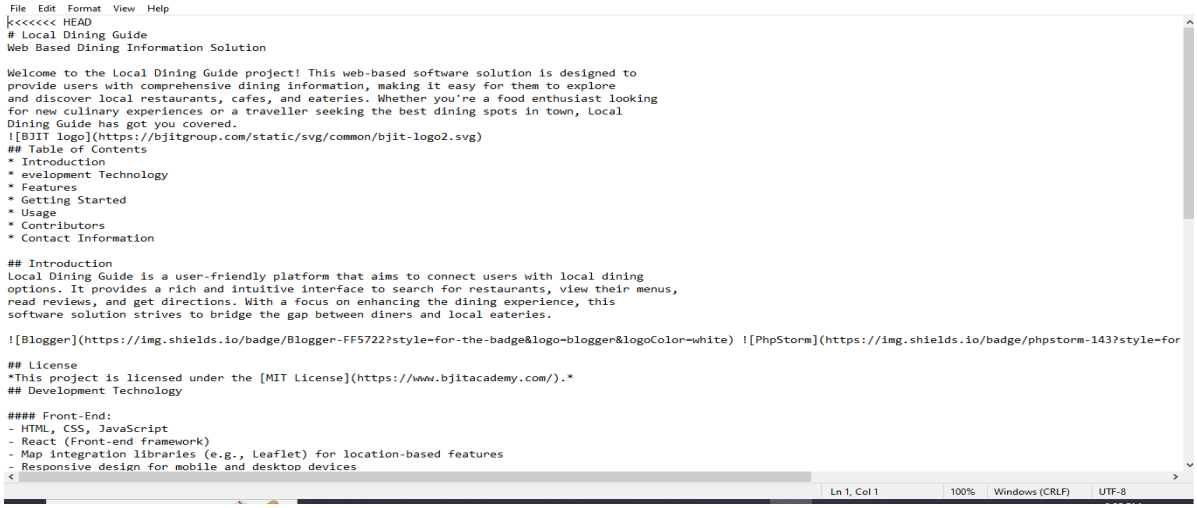
12. Pull the updated project file

```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (development)
$ git pull origin development
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 2), reused 6 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (12/12), 5.80 KiB | 110.00 KiB/s, done.
From github.com:dev-sharif-hasan/BJIT-Force
 * branch            development -> FETCH_HEAD
   905a5d9..41f4197  development -> origin/development
Updating 905a5d9..41f4197
Fast-forward
```

13. Select the specific commit

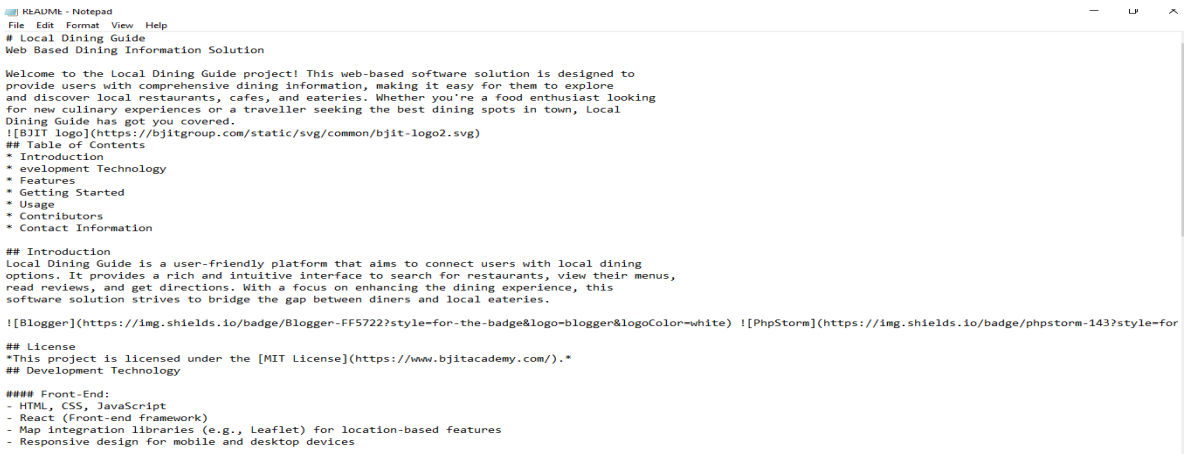
```
BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3)
$ git cherry-pick 9cf17af6cba8be2133bfccc1468a1080cacdf059
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 9cf17af... Doc(README.md): Adding prerequisite and installation
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
hint: Disable this message with "git config advice.mergeConflict false"

BJIT@DESKTOP-3N9TI6I MINGW64 /c/BJIT-Force (task3|CHERRY-PICKING)
$
```

14. Removing conflict from the README.me file



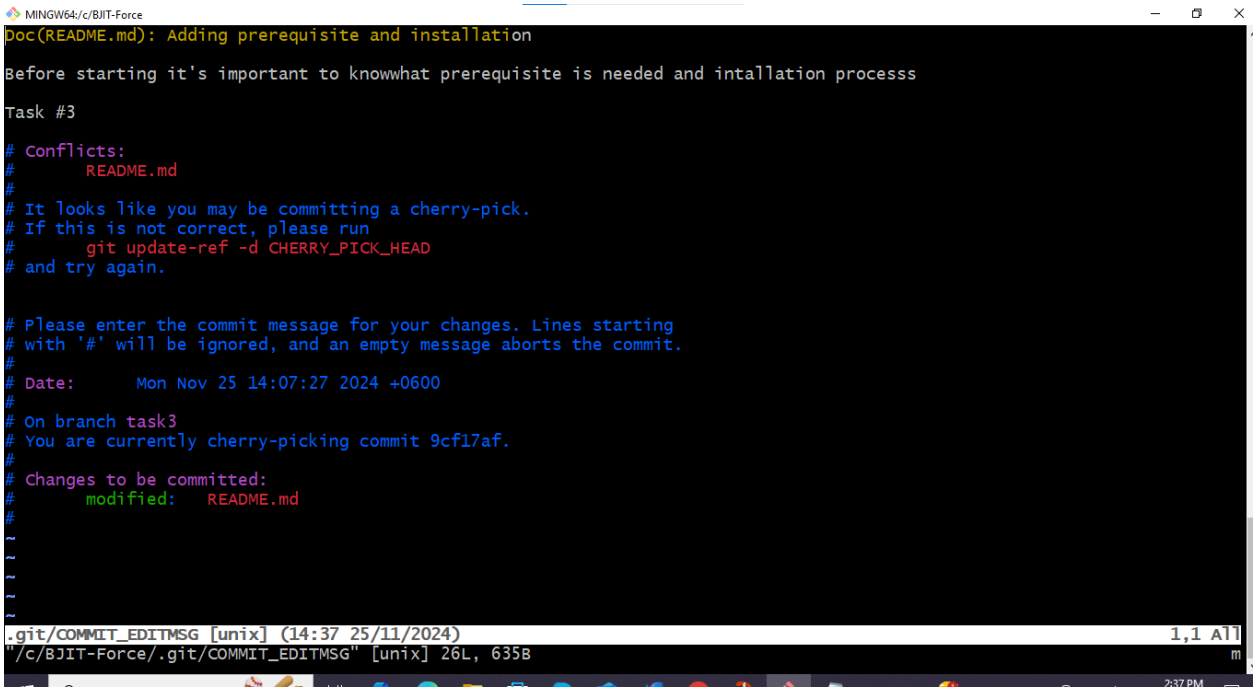
```
File  Edit  Format  View  Help
<<<<<< HEAD
# Local Dining Guide
Web Based Dining Information Solution

Welcome to the Local Dining Guide project! This web-based software solution is designed to
provide users with comprehensive dining information, making it easy for them to explore
and discover local restaurants, cafes, and eateries. Whether you're a food enthusiast looking
for new culinary experiences or a traveller seeking the best dining spots in town, Local
Dining Guide has got you covered.
![BJIT logo](https://bjitgroup.com/static/svg/common/bjit-logo2.svg)
## Table of Contents
* Introduction
* evelopment Technology
* Features
* Getting Started
* Usage
* Contributors
* Contact Information

## Introduction
Local Dining Guide is a user-friendly platform that aims to connect users with local dining
options. It provides a rich and intuitive interface to search for restaurants, view their menus,
read reviews, and get directions. With a focus on enhancing the dining experience, this
software solution strives to bridge the gap between diners and local eateries.

![Blogger](https://img.shields.io/badge/Blogger-FF5722?style=for-the-badge&logo=blogger&logoColor=white) ![PhpStorm](https://img.shields.io/badge/phpstorm-143?style=for

## License
*This project is licensed under the [MIT License](https://www.bjitacademy.com/).*
## Development Technology

#### Front-End:
- HTML, CSS, JavaScript
- React (Front-end framework)
- Map integration libraries (e.g., Leaflet) for location-based features
- Responsive design for mobile and desktop devices
```

15. After resolve the conflict

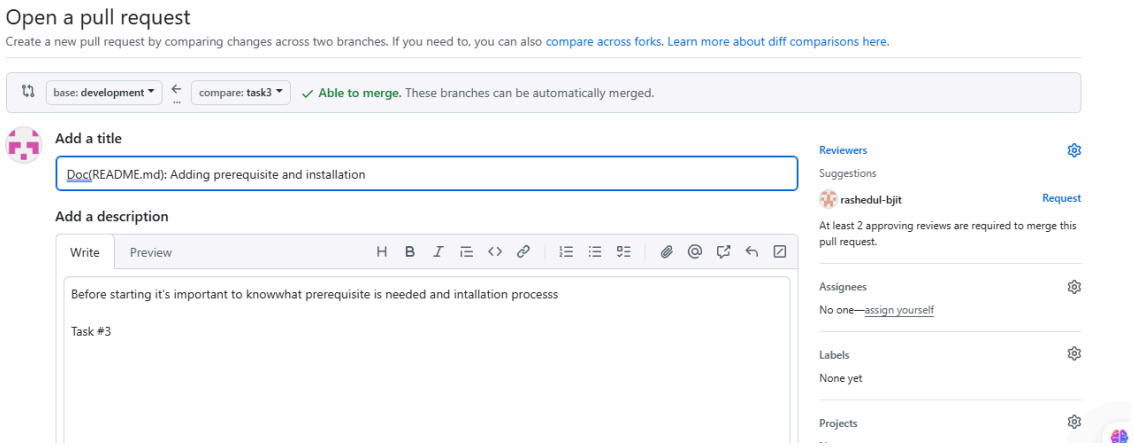


```
README - Notepad
File  Edit  Format  View  Help
# Local Dining Guide
Web Based Dining Information Solution

Welcome to the Local Dining Guide project! This web-based software solution is designed to
provide users with comprehensive dining information, making it easy for them to explore
and discover local restaurants, cafes, and eateries. Whether you're a food enthusiast looking
for new culinary experiences or a traveller seeking the best dining spots in town, Local
Dining Guide has got you covered.
![BJIT logo](https://bjitgroup.com/static/svg/common/bjit-logo2.svg)
## Table of Contents
* Introduction
* evelopment Technology
* Features
* Getting Started
* Usage
* Contributors
* Contact Information

## Introduction
Local Dining Guide is a user-friendly platform that aims to connect users with local dining
options. It provides a rich and intuitive interface to search for restaurants, view their menus,
read reviews, and get directions. With a focus on enhancing the dining experience, this
software solution strives to bridge the gap between diners and local eateries.

![Blogger](https://img.shields.io/badge/Blogger-FF5722?style=for-the-badge&logo=blogger&logoColor=white) ![PhpStorm](https://img.shields.io/badge/phpstorm-143?style=for

## License
*This project is licensed under the [MIT License](https://www.bjitacademy.com/).*
## Development Technology

#### Front-End:
- HTML, CSS, JavaScript
- React (Front-end framework)
- Map integration libraries (e.g., Leaflet) for location-based features
- Responsive design for mobile and desktop devices
```

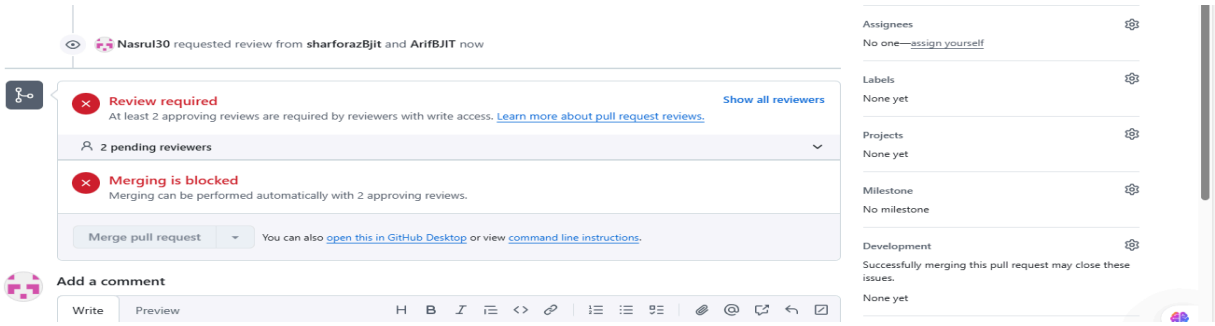
16. Write and save the resolve file after solving the conflict



17. Give the pull request for review

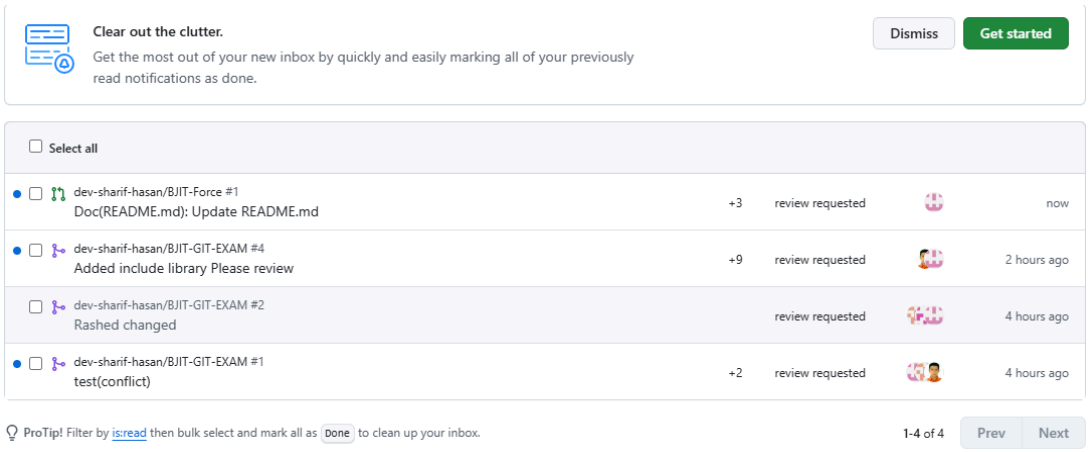


18. Review is pending

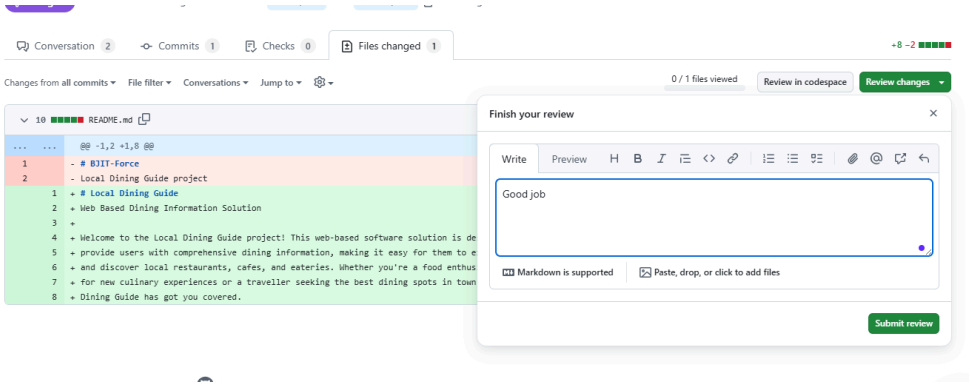


19. Review for others team-members

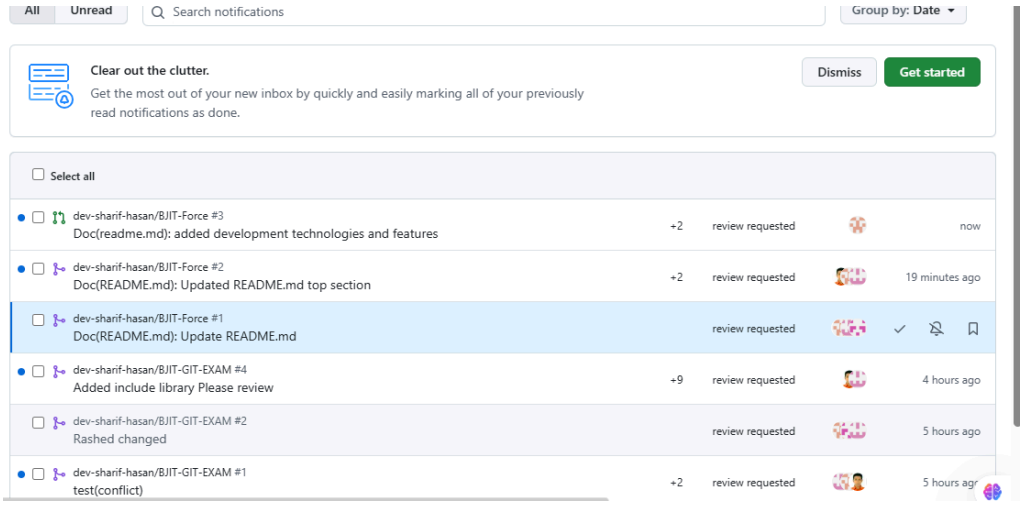
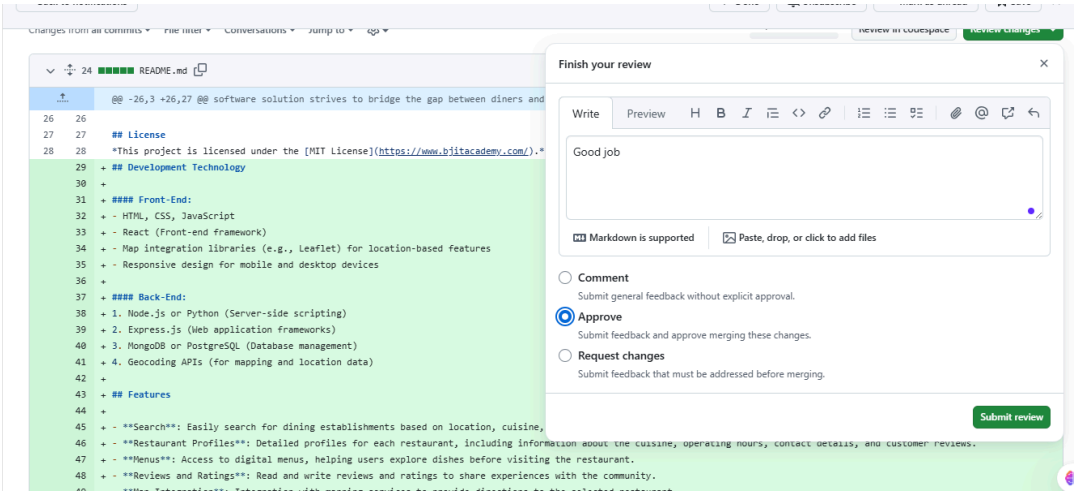
a) Give the review to the team-mate sharif



Review process



b) Give the review to the team mate Md. Rashedul Islam



Review process: