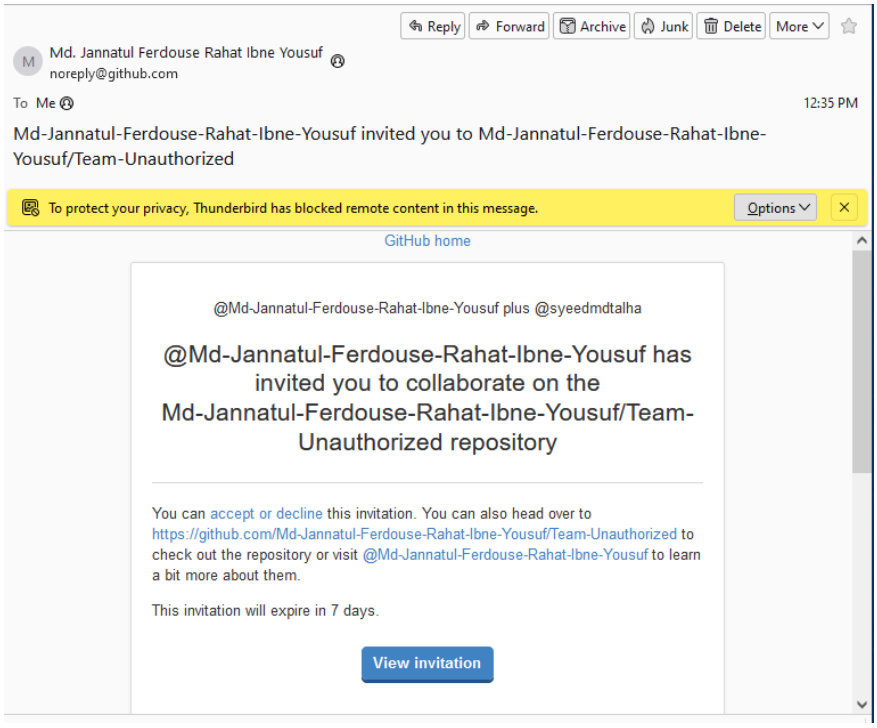
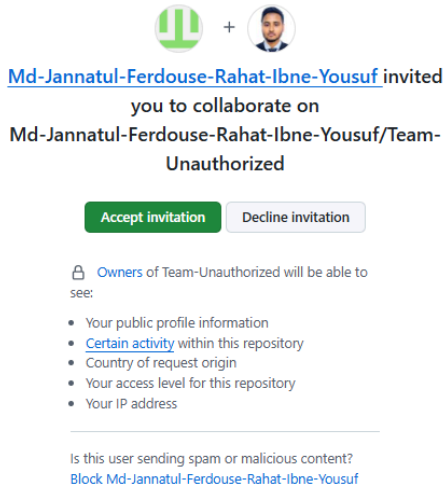
Collaborating on a GitHub Repository

I received an invitation to join a GitHub repository as a collaborator.



After reviewing the invite, I accepted it through the notification on GitHub. This granted me access to the repository, allowing me to contribute directly to the project.



Verifying SSH Connection and Git Configuration

To ensure smooth collaboration with GitHub, I tested my SSH connection by running the ssh -T git@github.com command. It confirmed that my SSH key was successfully linked to my GitHub account.

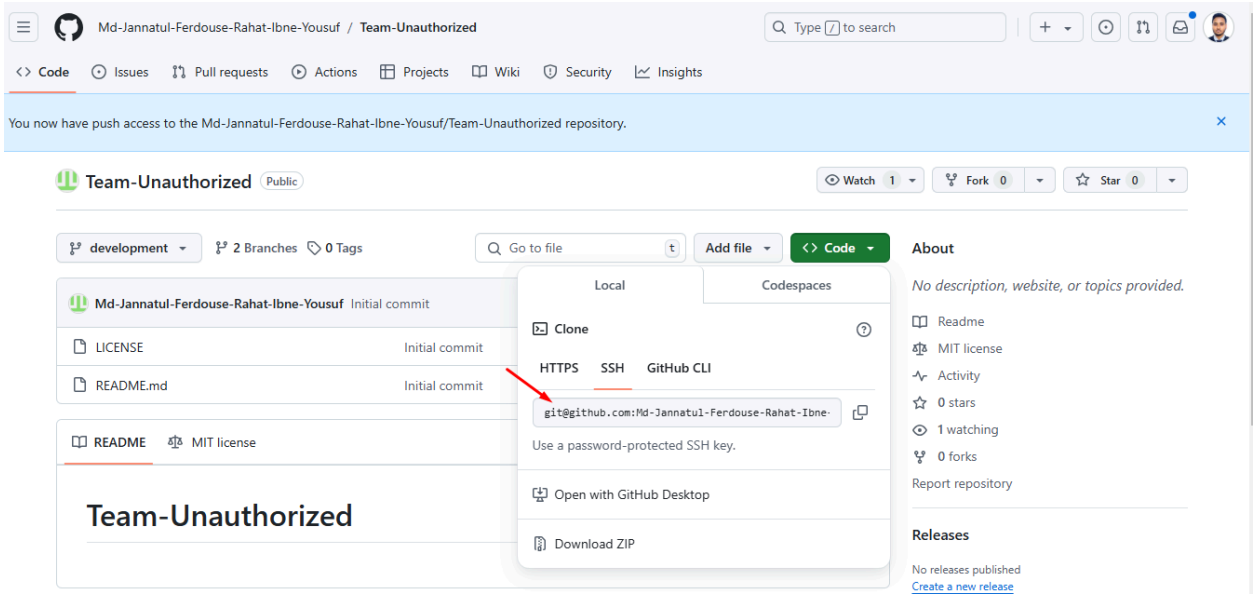
```
MINGW64:/c/Users/BJIT/Desktop/Git_Exame

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame
$ ssh -T git@github.com
Hi syeedmdtalha! You've successfully authenticated, but GitHub
 does not provide shell access.
```

Next, I checked my Git configuration using git config --list. This displayed my username and email, which I had previously set using:
git config --global user.name "YourUsername"
git config --global user.email "youremail@example.com"
Both were correctly configured, confirming everything was set up properly.

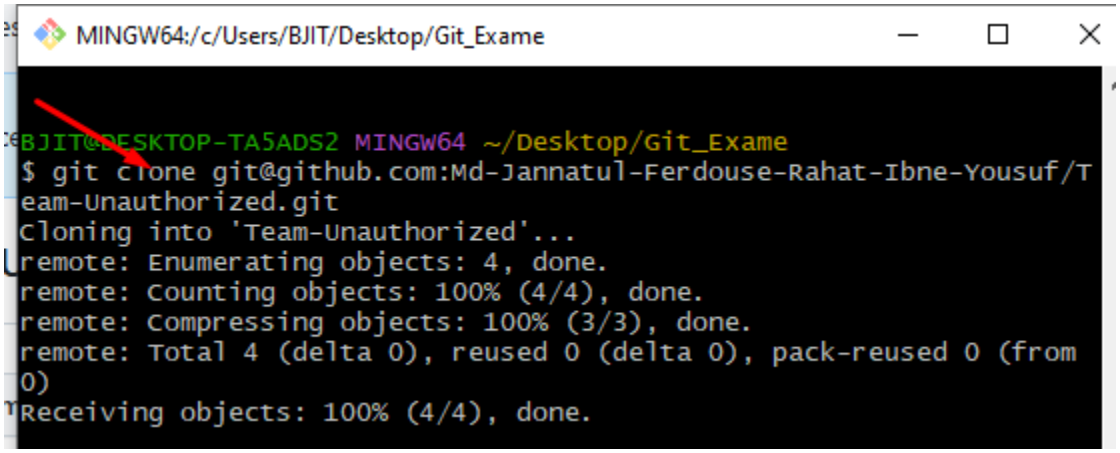
```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca
-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Syeed MD Talha
user.email=syeed.talha@bjitacademy.com
```

Copying SSH URL and Cloning a Git Repository

I navigated to the GitHub repository and copied the SSH URL from the "Code" dropdown menu. Using this URL, I cloned the repository to my local machine with the command:
git clone git@github.com:username/repository.git.



This created a local copy of the repository, allowing me to work on it directly.



Accessing the Repository and Viewing Branches

After cloning the repository, I navigated into its directory using cd repository-name.

```
MINGW64:/c/Users/BJIT/Desktop/Git_Exame/Team-Unauthorized

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame
$ cd Team-Unauthorized

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized
(development)
$ git branch
* development

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized
(development)
$
```

Creating a New Branch

Inside the repository, I created a new branch named `task3branch` using the command: `git branch task3branch`.

To start working on it, I switched to the branch with: `git checkout task3branch`.

This allowed me to isolate my changes from the main branch.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized
(development)
$ git checkout -b task3branch
Switched to a new branch 'task3branch'

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized
(task3branch)
$ |
```

Updating the README File

While on the `task3branch`, I opened the `README.md` file and added some text describing the purpose of the project.

README.md

C: > Users > BJIT > Desktop > Git_Exame > Team-Unauthorized > README.md > # Getting Started > ## Installation

```md
# Getting Started

_Follow these steps to get Local Dining Guide up and running on your local machine._

## Prerequisites

**Web Browser:** Ensure you have a modern web browser installed to access the Local Dining Guide interface.

</br>

**Internet Connection:** A stable internet connection is required to fetch restaurant data and images.

</br>


## Installation

1. **Clone the Repository:**
   Start by cloning this repository to your local machine.
   ```bash
   git clone https://github.com/yourusername/local-dining-guide.git
   ```

2. **Navigate to Directory:**
   Move into the project directory
```

Checking Git Status and Staging a File

I checked the repository status using `git status`. It showed the `README.md` file as modified and unstaged.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized
(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 direct
ory)
        modified:   README.md

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

Next, I staged the file with `git add README.md` and ran `git status` again. This time, the file appeared under the "Changes to be committed" section, indicating it was successfully staged and ready for commit.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized
(task3branch)
$ git add README.md

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3bra
nch)
$ git status
On branch task3branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```

Committing Changes to `task3branch` and Checking Last Commit

After staging the `README.md` file, I committed the changes to the `task3branch` with the following command:
`git commit -m "Updated README with project details"`.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3branch)
$ git commit -m"docs(readme): update installation instructions" -m"Refs.#13125"
[task3branch 5ef8e46] docs(readme): update installation instructions
 1 file changed, 34 insertions(+), 1 deletion(-)
```

To check the last commit message, I used the command: `git log -1`.

This displayed the most recent commit message, confirming that the changes were successfully committed.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3branch)
$ git log -1
commit 5ef8e4661841971592df7933933b3afcf2dce63f (HEAD -> task3branch)
Author: Syeed MD Talha <syeed.talha@bjitacademy.com>
Date:   Mon Nov 25 14:41:48 2024 +0600

    docs(readme): update installation instructions

    Refs.#13125
```

Retrieving the Commit ID

To get the commit ID for the most recent commit, I used the command:
`git log -1 --oneline`.

This displayed the commit ID along with the commit message in a concise format, allowing me to easily copy the commit ID for reference.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3branch)
$ git log --oneline
5ef8e46 (HEAD -> task3branch) docs(readme): update installation instructions
1f6f44d (origin/main, origin/development, origin/HEAD, development) Initial commit
```

After that i checked out to the development branch

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3branch)
$ git checkout development
Switched to branch 'development'
Your branch is up to date with 'origin/development'.

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (development)
$ |
```

Pulling Changes from the Development Branch

To ensure my `task3branch` is up to date with the latest changes from the `development` branch.. Then, I pulled the latest changes from the `development` branch using the following command:
`git pull origin development`.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (development)
$ git pull origin development
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 20 (delta 4), reused 14 (delta 3), pack-reused 0 (from 0)
Unpacking objects: 100% (20/20), 7.28 KiB | 109.00 KiB/s, done.
From github.com:Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf/Team-Unauthorized
 * branch            development -> FETCH_HEAD
   1f6f44d..771222f  development -> origin/development
Updating 1f6f44d..771222f
Fast-forward
 README.md | 91 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 1 file changed, 90 insertions(+), 1 deletion(-)

BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (development)
$
```

Creating a New Branch `task3branch_copy`

After pulling the latest changes from the `development` branch, I created a new branch called `task3branch_copy` by running:
`git checkout -b task3branch_copy`.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/(
y)
$ git checkout -b task3branch_copy
```

Using `git cherry-pick` with a Commit ID

To apply a specific commit from another branch (such as from `task3branch`) to my new branch `task3branch_copy`, I used the `git cherry-pick` command with the commit ID. The command was:
`git cherry-pick 5ef8e46`

This applied the changes from the specified commit to the `task3branch_copy`, allowing me to incorporate selected changes without merging the entire branch.

```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3branch_cop
y)
$ git cherry-pick 5ef8e46
```

Resolving Merge Conflicts Manually

After running the `git cherry-pick` command, I encountered a merge conflict. To resolve it manually:

1. I opened the conflicting files and identified the sections marked with `<<<<<<<`, `=======`, and `>>>>>>>`. These markers showed the conflicting changes from both branches.
2. I edited the file to resolve the conflict by choosing the correct changes or merging them as needed.
3. After resolving the conflict, I removed the conflict markers and saved the file.

To finalize the resolution, I staged the resolved files using:
`git add .`

Then, I completed the cherry-pick process by committing the resolved conflict with:
`git commit`.

C: > Users > BJIT > Desktop > Git_Exame > Team-Unauthorized > README.md > # Local Dining Guide > ## Getting Started > ### Installation

```md
# Local Dining Guide

Web Based Dining Information Solution

<br>
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.


## Development Technology

#### **Front-End**
- HTML, CSS, JavaScript
- React (Front-end framework.)
- Map Integration Libraries (e.g., Leaflet) for location-based features.
- Responsive Design Optimized for mobile and desktop devices.

#### **Back-End**
1. Node.js or Python (Server-side scripting.)
2. Express.js (Web application framework.)
3. MongoDB or PostgreSQL (Database management systems.)
4. Geocoding APIs (For mapping and location data.)
```

This committed the conflict resolution, and the changes were successfully integrated into `task3branch_copy`.

Pushing Changes to the Remote Repository

After resolving the merge conflict and committing the changes in `task3branch_copy` i push to the remote repository.

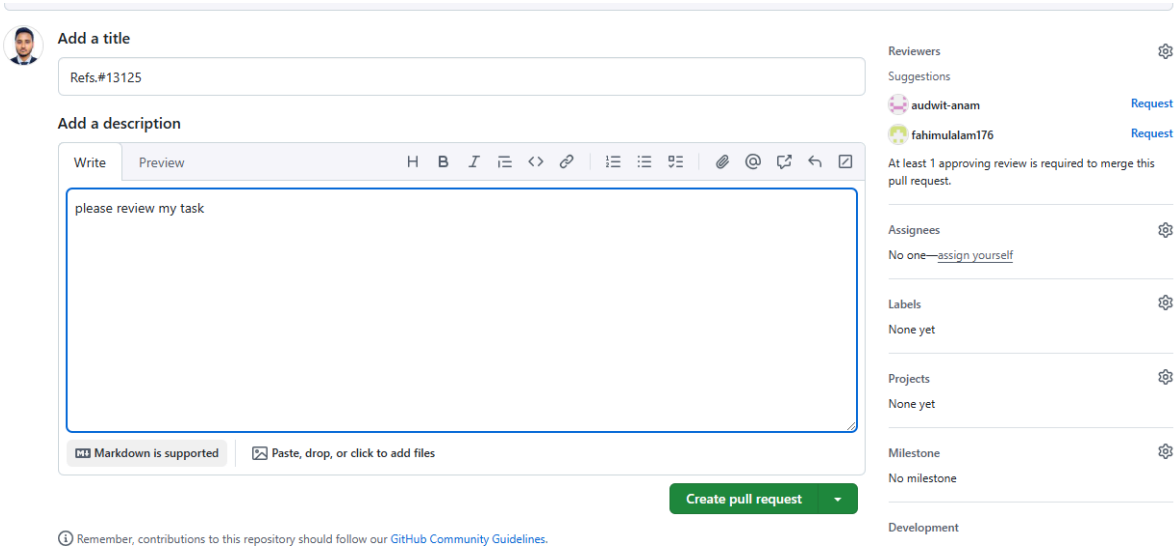
```
BJIT@DESKTOP-TA5ADS2 MINGW64 ~/Desktop/Git_Exame/Team-Unauthorized (task3branch_cop
y)
$ git push -u origin task3branch_copy
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 754 bytes | 377.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'task3branch_copy' on GitHub by visiting:
remote:      https://github.com/Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf/Team-Unautho
rized/pull/new/task3branch_copy
remote:
To github.com:Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf/Team-Unauthorized.git
 * [new branch]      task3branch_copy -> task3branch_copy
branch 'task3branch_copy' set up to track 'origin/task3branch_copy'.
```

Creating a Pull Request

After pushing the changes to the `main` branch, I created a pull request to merge the updates from `task3branch_copy` into the `main` branch on GitHub. Here are the steps I followed:

1. I went to the repository on GitHub.
2. I clicked on the **"Pull Requests"** tab.
3. Then, I clicked on the **"New Pull Request"** button.
4. I selected `task3branch_copy` as the source branch and `main` as the target branch.
5. I added a title and description for the pull request, explaining the changes I made.
6. Finally, I clicked **"Create Pull Request"** to submit it for review.

Now, the pull request was ready for review and merging.



Pull Request Accepted by Team Leader

After I created the pull request, my team leader reviewed the changes and accepted it. Once the pull request was approved, they merged it into the `main` branch on GitHub. This successfully incorporated my changes into the project, and the `main` branch was updated with the latest contributions from `task3branch_copy`.

Refs.#13125 #4

Edit | Code

Merged | Md-Jannatul-Fer... merged 1 commit into `development` from `task3branch_copy` 1 minute ago

Conversation 0 | Commits 1 | Checks 0 | Files changed 1 | +39 −0

syeedmdtalha commented 2 minutes ago — Collaborator

please review my task

`Refs.#13125` 7f44ec8

Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf approved these changes 1 minute ago — View reviewed changes

Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf left a comment — Owner

Looks good.

Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf merged commit **9ee5cd2** into `development` 1 minute ago — Revert

Reviewers
Md-Jannatul-Ferdouse-Rahat-Ibne-Yousuf

Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone