

# Git and GitHub Practical Exam

## Group 4

ID	29
Name	Shahadat Hossain
Task No	2
Task Description	Add Development Technology and Features section.

### Activities

**Step 1:** Clone remote repository.

```
user@DESKTOP-6FCER5A MINGW64 /h/project
$ git clone git@github.com:arif-rahman-cse/group_4.git
Cloning into 'group_4'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

**Step 2:** Navigating to the group\_4 folder.

```
user@DESKTOP-6FCER5A MINGW64 /h/project
$ cd group_4/

user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (development)
$ |
```

**Step 3:** Pulling development branch for latest changes.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (development)
$ git pull origin development
From github.com:arif-rahman-cse/group_4
* branch      development -> FETCH_HEAD
Already up to date.
```

**Step 4:** Opening code editor in the current folder.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (development)
$ code .

user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (development)
$ |
```

**Step 5:** Create feature/add-technology-features-section branch.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (development)
$ git branch feature/add-technology-features-section
```

**Step 6:** Use git checkout command to navigate to feature/add-technology-features-section branch.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (development)
$ git checkout feature/add-technology-features-section
Switched to branch 'feature/add-technology-features-section'
```

**Step 7:** Add **Development Technology** section in the Readme file and then check status.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git status
On branch feature/add-technology-features-section
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")
```

**Step 8:** Use 'git add .' command to stage all changes and then check status.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git add .

user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git status
On branch feature/add-technology-features-section
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
```

**Step 9:** Use 'git commit' command to commit the current stage changes by following the Conventional Commits rules.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git commit -m "docs(readme): add development technology section to README"
>
> This commit adds a new section to the README file providing details about the front-end and back-end technologies used in the project. It includes information about HTML, CSS, JavaScript, React, map integration libraries like Leaflet, and the back-end technologies such as Node.js, Express.js, and databases like MongoDB or PostgreSQL. Additionally, geocoding APIs are mentioned for mapping and location data.
>
> This information will help developers and contributors understand the technology stack used in the project."
(feature/add-technology-features-section 2da8bd8) docs(readme): add development technology section to README
1 file changed, 15 insertions(+), 1 deletion(-)
```

**Step 10:** Add **Features** section in the Readme file and then check status.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git status
On branch feature/add-technology-features-section
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")
```

**Step 11:** Use 'git add .' command to stage all changes and then check status.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git add .

user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git status
On branch feature/add-technology-features-section
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
```

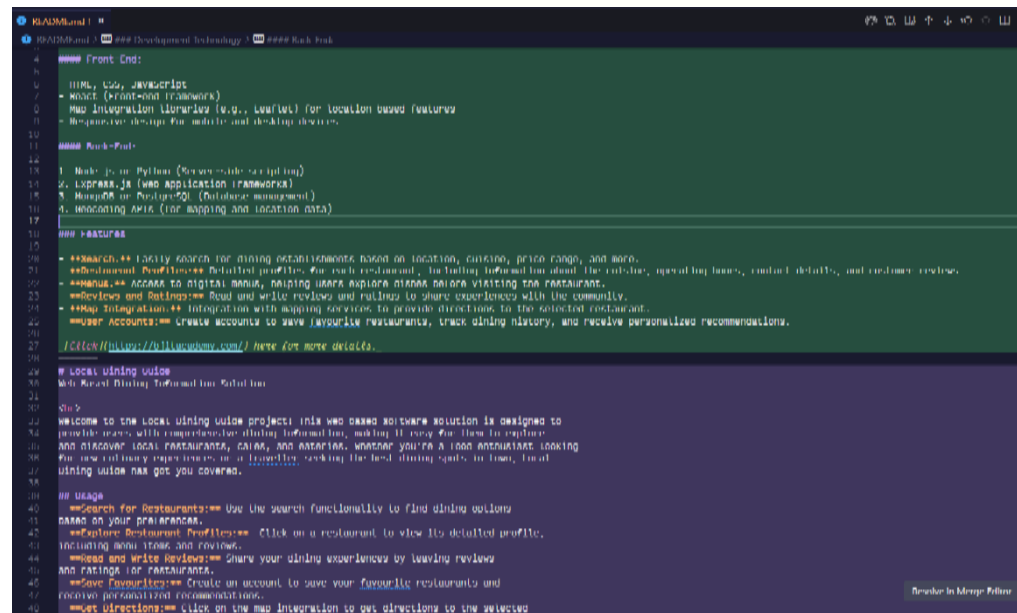
**Step 12:** Use 'git commit' command to commit the current stage changes by following the Conventional Commits rules (second commit).

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git commit -m "docs(readme): add features section to README
>
> Provide details about key features such as search functionality, restaurant profiles, menus, reviews and ratings, map integration, and user accounts. Include a link for more details."
(feature/add-technology-features-section 85ee58a) docs(readme): add features section to README
1 file changed, 11 insertions(+)
```

**Step 13:** Use 'git pull origin' command to get the latest changes.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section|MERGING)
$ git pull origin development
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 1), reused 6 (delta 1), pack-reused 0
Unpacking objects: 100% (7/7), 1.84 KiB | 5.00 KiB/s, done.
From github.com:arif-rahman-cse/group_4
* branch                development -> FETCH_HEAD
c398e94..c014101 development -> origin/development
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

**Step 14:** Fix conflict and marge the commit.



```
10 README.md
11 ### Front End:
12 1. HTML, CSS, JavaScript
13 2. React (Front-end framework)
14 3. Map Integration Libraries (e.g., Leaflet) for location-based features
15 4. Responsive design for mobile and desktop devices.
16
17 ### Back-End:
18 1. Node.js or Python (Backend scripting)
19 2. Express.js (Web application framework)
20 3. MongoDB or PostgreSQL (Database management)
21 4. RESTful APIs (for mapping and location data)
22
23 ### Features
24
25 - **Search:** Easily search for dining establishments based on location, cuisine, price range, and more.
26 - **Restaurant Profiles:** Detailed profiles for each restaurant, including information about the cuisine, operating hours, contact details, and customer reviews.
27 - **Menus:** Access to digital menus, helping users explore dining options without visiting the restaurant.
28 - **Reviews and Ratings:** Read and write reviews and ratings to share experiences with the community.
29 - **Map Integration:** Integration with mapping services to provide directions to the selected restaurant.
30 - **User Accounts:** Create accounts to save favorite restaurants, track dining history, and receive personalized recommendations.
31
32 [Click here for more details.]
33
34 # Local Dining Guide
35 Welcome to the Local Dining Guide! 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 traveler seeking the best dining spots in town, Local Dining Guide has got you covered.
36
37 ## Usage
38
39 - **Search for Restaurants:** Use the search functionality to find dining options based on your preferences.
40 - **Explore Restaurant Profiles:** Click on a restaurant to view its detailed profile, including menu items and reviews.
41 - **Read and Write Reviews:** Share your dining experiences by leaving reviews and ratings for restaurants.
42 - **Save Favorites:** Create an account to save your favorite restaurants and receive personalized recommendations.
43 - **Get Directions:** Click on the map integration to get directions to the selected
```

**Step 15:** Push the changes in the remote branch.

```
user@DESKTOP-6FCER5A MINGW64 /h/project/group_4 (feature/add-technology-features-section)
$ git push --set-upstream origin feature/add-technology-features-section
```

