

COURSE DETAILS

| | | | |
|--------------------------------|---|-------------------------------------|----------|
| Academic Year | : 2024 – 25 (Odd Sem) | Year of Study & Semester | : I & 01 |
| Course Code & Title | : U23CS381 & Application Design and Development | | |

Guidelines for Writing Git Commit Messages

☀ **Attention, Future Superstars!** ☀

1. Header:

The header is the most crucial part of a commit message. It includes the type of change, an optional scope, and a brief subject.

- **Type:** This indicates the nature of the change. Some common types include:
 - **feat**: A new feature.
 - **fix**: A bug fix.
 - **docs**: Documentation changes.
 - **style**: Code style changes (formatting, missing semi-colons, etc.).
 - **refactor**: Code refactoring without changing external behavior.
 - **test**: Adding or fixing tests.
 - **chore**: Maintenance tasks (e.g., build process changes).
- **Scope:** The scope is optional but helps indicate the area of the codebase affected by the change. It's written in parentheses after the type:
 - **feat(auth)**: A new feature related to authentication.
 - **fix(ui)**: A bug fix in the user interface.
- **Subject:** This is a brief description of the change. Write it in the imperative mood (as if giving a command):
 - **Good**: Add search functionality to the header component
 - **Bad**: Added search functionality (Avoid past tense)

Example Header:

```
feat(auth): add password reset feature
```

2. Body:

The body provides a detailed explanation of the change. It includes what was changed, why it was changed, and any additional context or background.

- **What:** Describe what has been done in this commit.
 - Example: - Added password reset functionality to the authentication system
- **Why:** Explain the reasons behind making this change.
 - Example: - This change allows users to reset their passwords if they forget them, improving user experience and security.
- **How:** Briefly describe how the change was implemented.

- Example: - Implemented backend API for sending password reset emails and frontend forms for users to submit their new passwords.

Example Body:

- Added password reset functionality to the authentication system
- This change allows users to reset their passwords if they forget them, improving user experience and security.
- Implemented backend API for sending password reset emails and frontend forms for users to submit their new passwords.

3. Footer:

The footer is used for referencing any issues related to the commit or providing additional information.

- **Issue Reference:** If the commit resolves an issue, include the reference.
 - Example: Closes #123
- **Example Body:**

Closes #123

Full Example Commit Message:

feat(auth): add password reset feature

- Added password reset functionality to the authentication system
- This change allows users to reset their passwords if they forget them, improving user experience and security.
- Implemented backend API for sending password reset emails and frontend forms for users to submit their new passwords.

Closes #123

Additional Tips for Writing Good Commit Messages:

1. **Be Concise but Descriptive:** The header should summarize the change in about 50 characters or less.
2. **Use the Imperative Mood:** Write as if you are commanding the code to do something. For example, "Add feature" instead of "Added feature."
3. **Separate Logical Changes:** Each commit should represent a single logical change. Avoid bundling unrelated changes in one commit.
4. **Reference Issues and PRs:** Use the footer to reference issues, pull requests, or other related tickets to provide context.

-X-X-X-X-

☀️ Tech Wizards! ☀️

