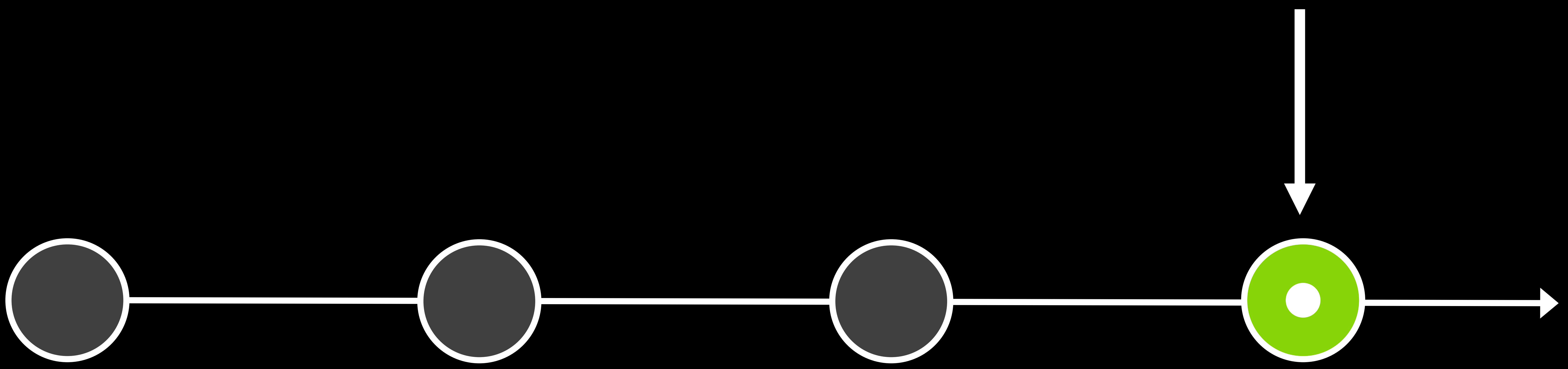
5 tips

Clean

Commits



git + GitHub

Nabi Karampour
@thisisnabi

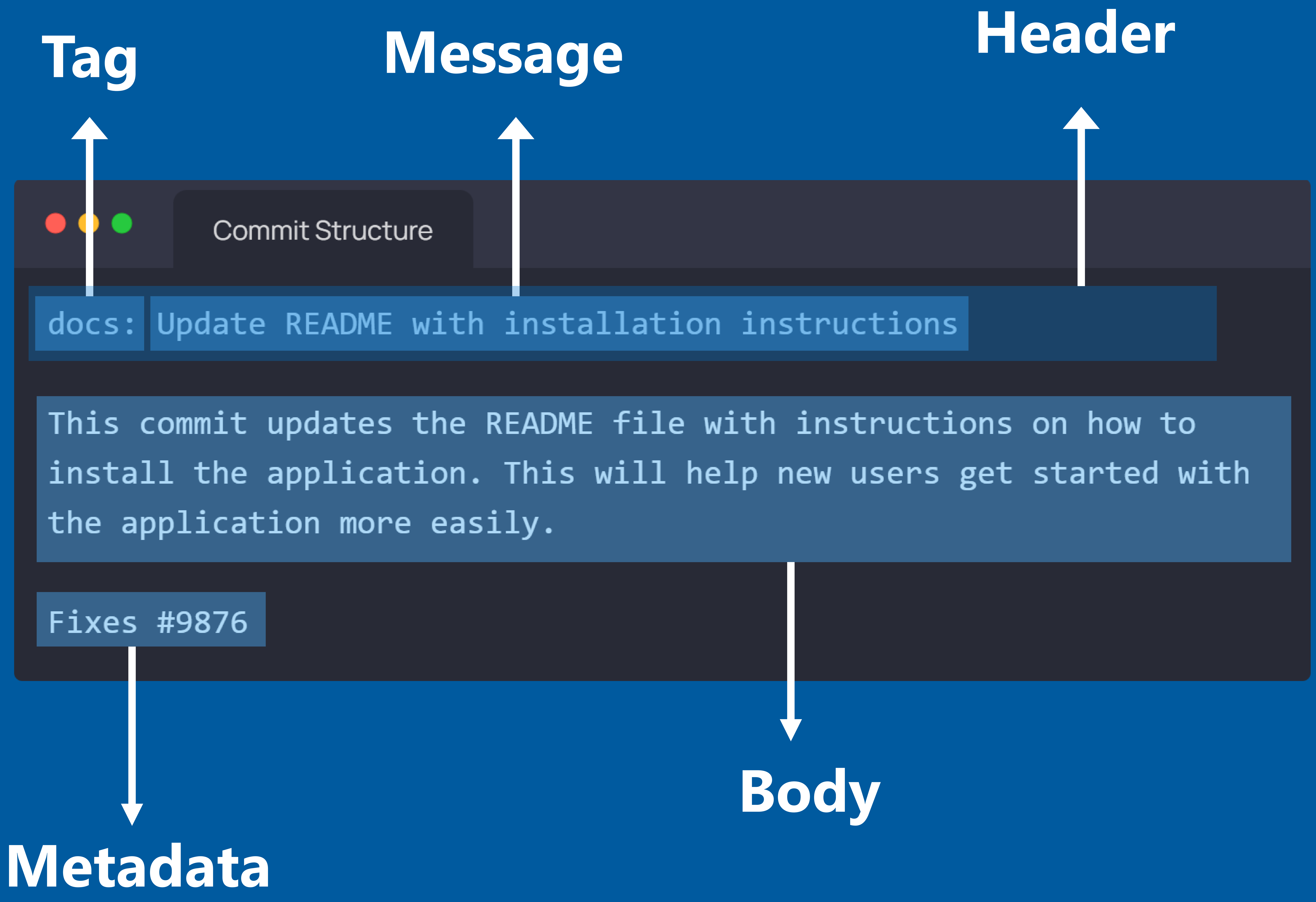
Why it's important?

Clean commits with clear and descriptive messages

enables easier collaboration among developers,

and simplify the process of debugging and maintaining codebases.

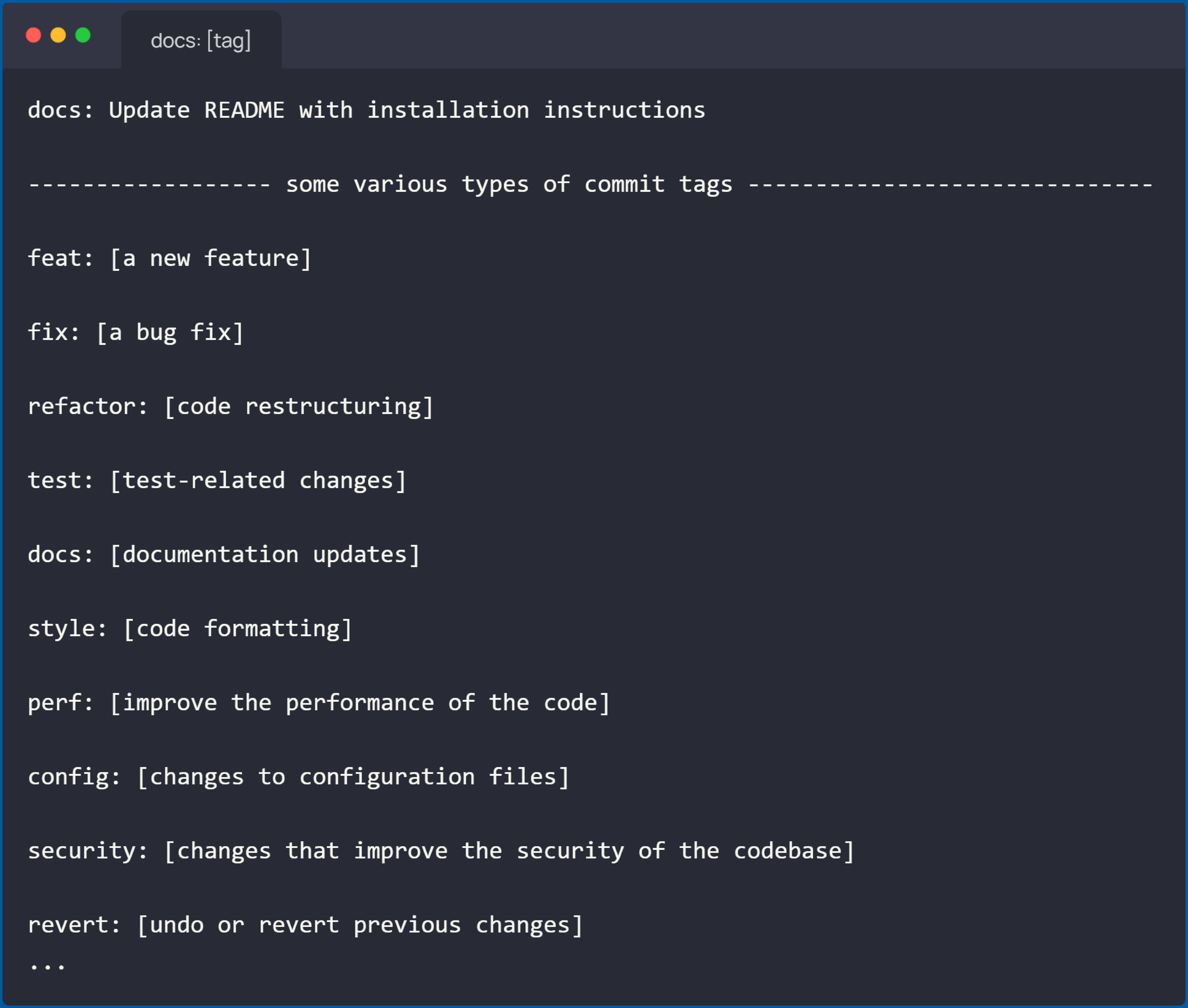
Commit Structure



```
docs: Update README with installation instructions

This commit updates the README file with instructions on how to
install the application. This will help new users get started with
the application more easily.

Fixes #9876
```

1 Use Commit Tag

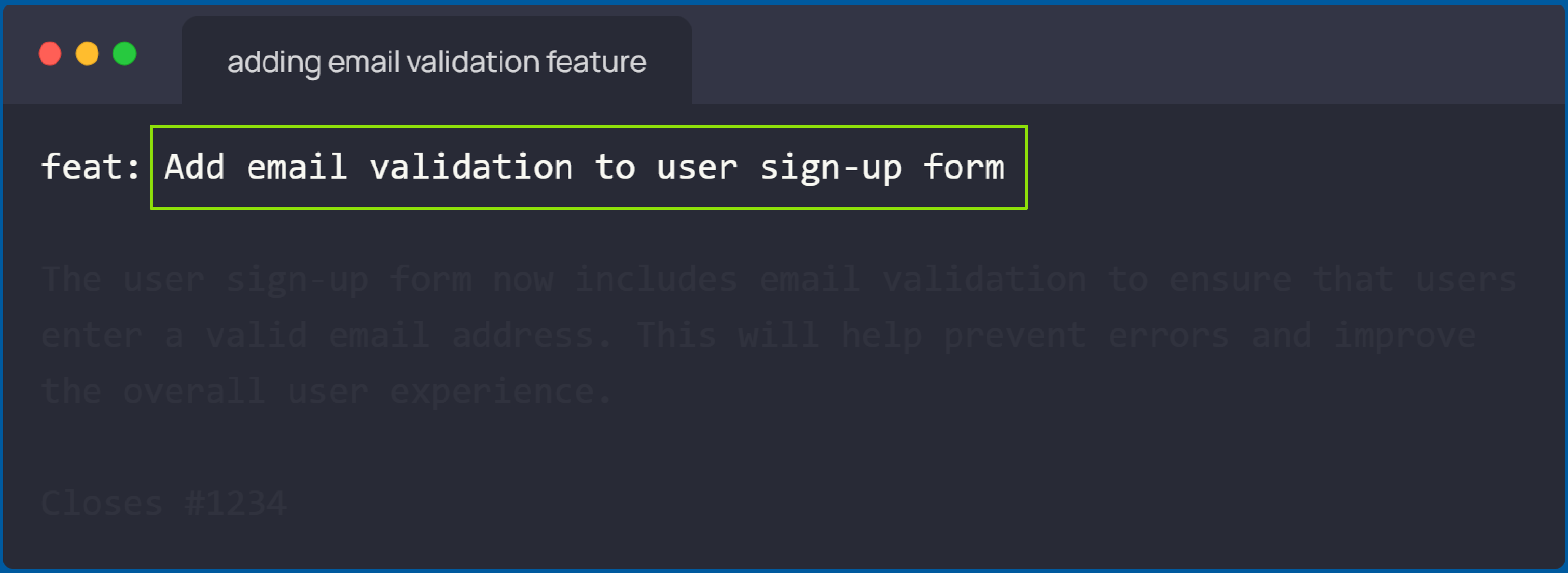
Tag in Commits indicates the type of change made.



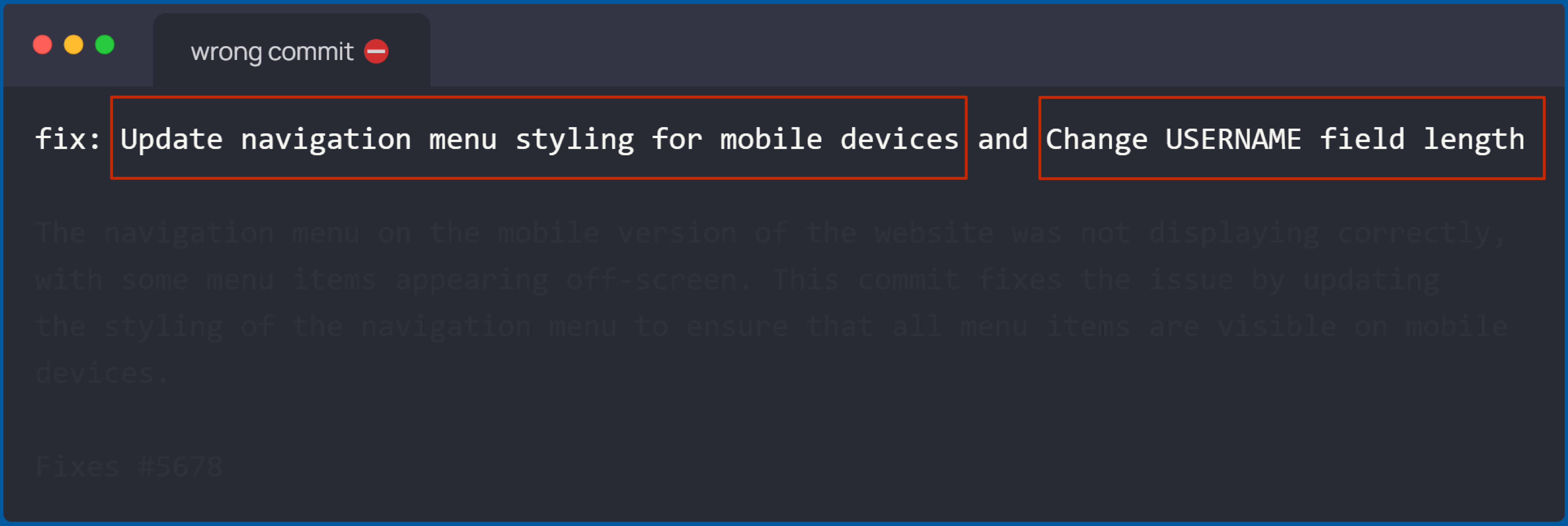
```
docs: Update README with installation instructions

------------------ some various types of commit tags ------------------------------

feat: [a new feature]

fix: [a bug fix]

refactor: [code restructuring]

test: [test-related changes]

docs: [documentation updates]

style: [code formatting]

perf: [improve the performance of the code]

config: [changes to configuration files]

security: [changes that improve the security of the codebase]

revert: [undo or revert previous changes]
...
```

2 Keep it simple and focused

Each commit should represent a single logical change.



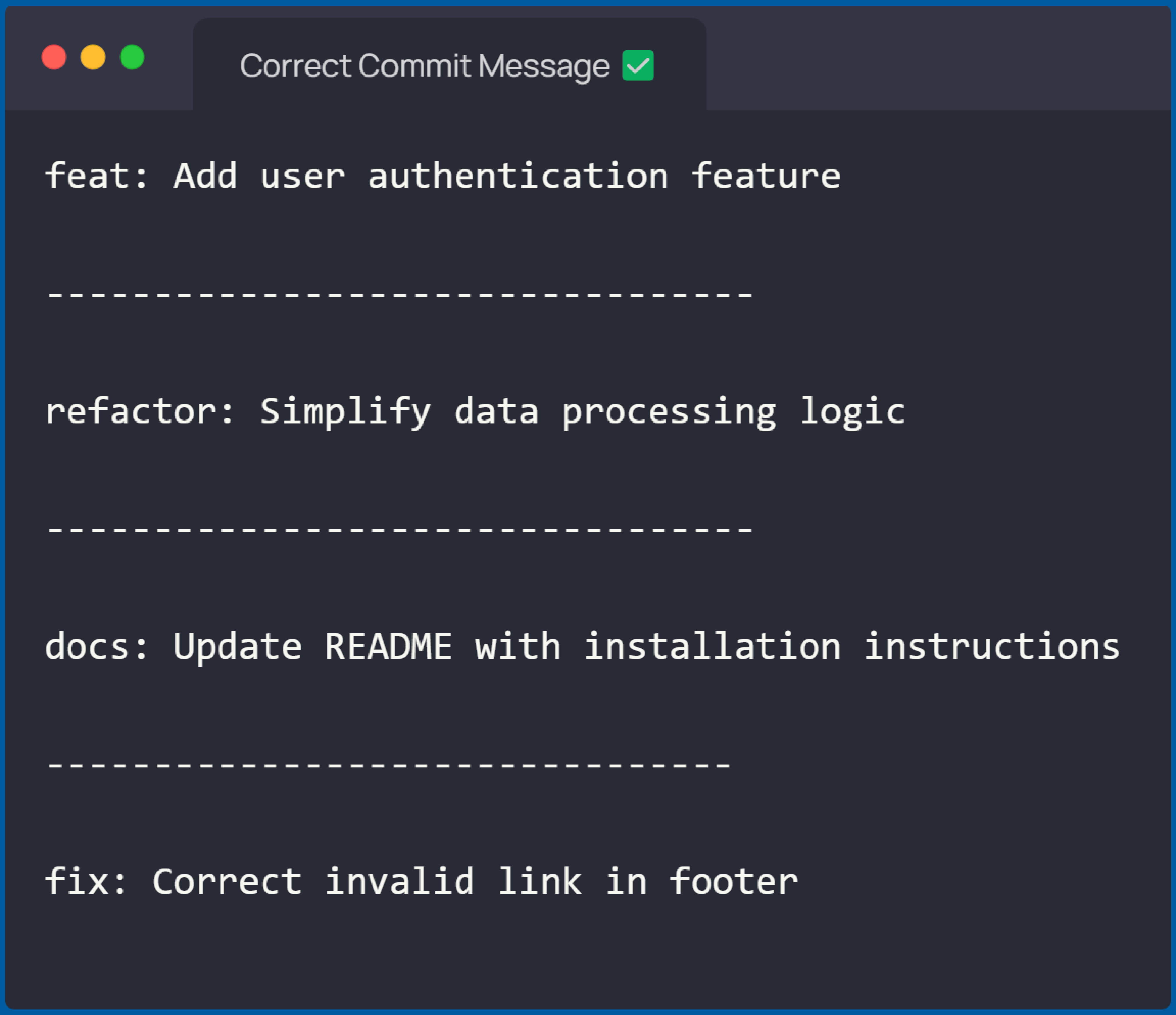
```
feat: Add email validation to user sign-up form

The user sign-up form now includes email validation to ensure that users
enter a valid email address. This will help prevent errors and improve
the overall user experience.

Closes #1234
```



```
fix: Update navigation menu styling for mobile devices and Change USERNAME field length

The navigation menu on the mobile version of the website was not displaying correctly,
with some menu items appearing off-screen. This commit fixes the issue by updating
the styling of the navigation menu to ensure that all menu items are visible on mobile
devices.

Fixes #5678
```

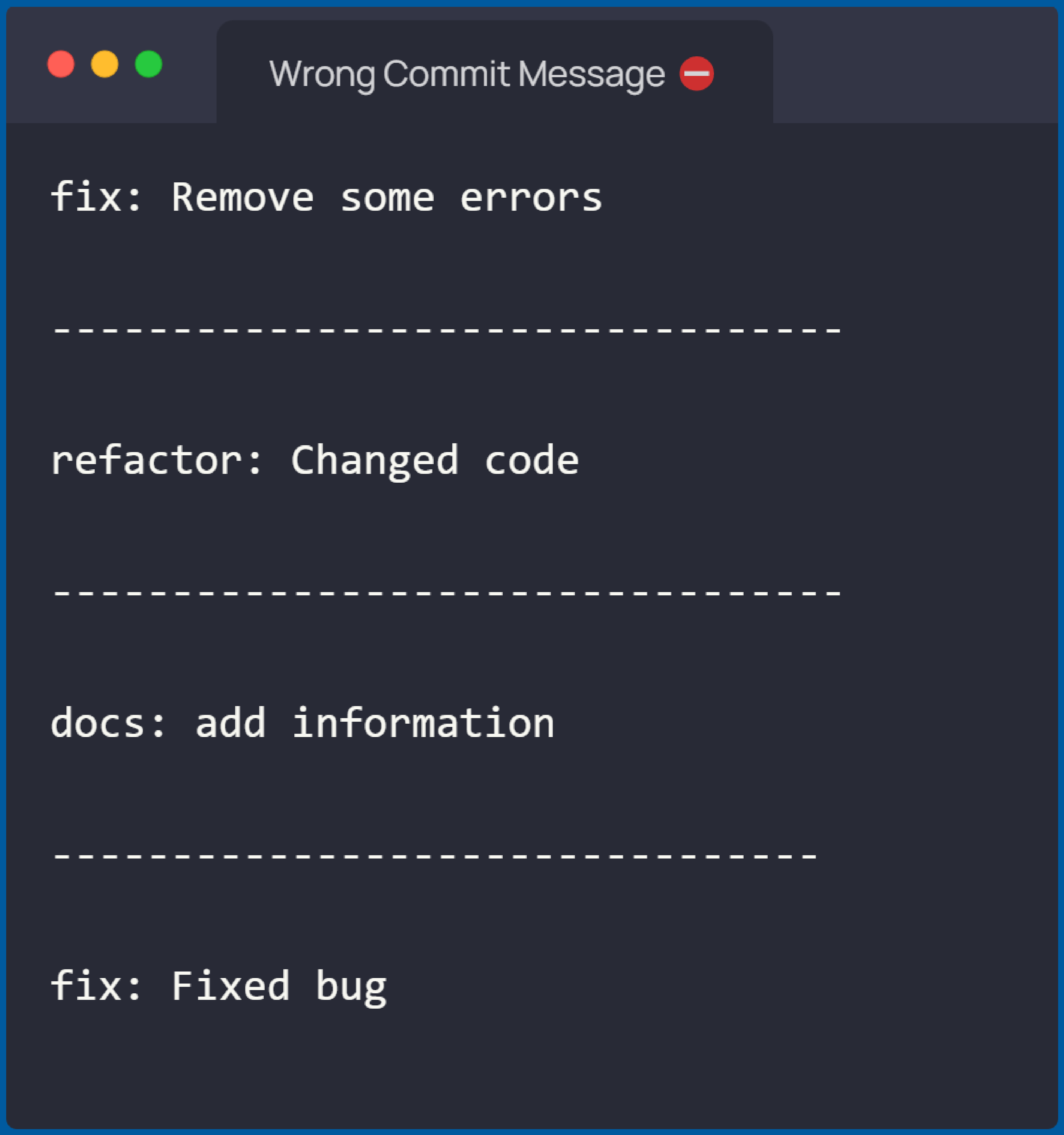
3 clear and descriptive message

The commit message should summarize the changes you made in the commit. It should be brief and descriptive.



Correct Commit Message ✅

```
feat: Add user authentication feature

---------------------------------

refactor: Simplify data processing logic

---------------------------------

docs: Update README with installation instructions

--------------------------------

fix: Correct invalid link in footer
```



Wrong Commit Message ⛔

```
fix: Remove some errors

---------------------------------

refactor: Changed code

---------------------------------

docs: add information

--------------------------------

fix: Fixed bug
```

⚠ It should start with a verb in the present tense.

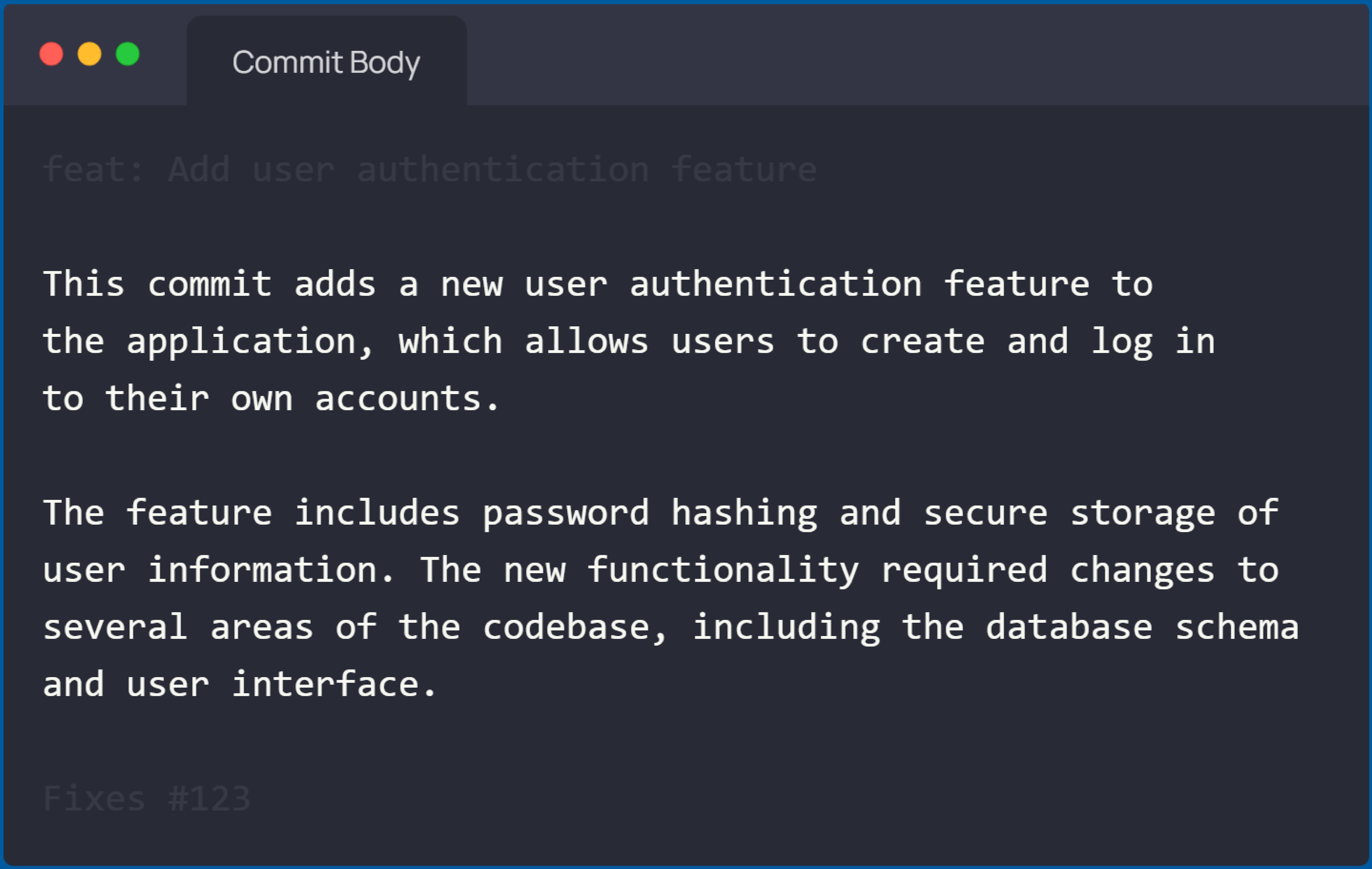
+ Fix typo in README.md
+ Add new feature to homepage.
- Adding new ...
- Fixed typo ...

⚠ Keep the commit message under 72 characters.

+ Proper format for correct display in log/tools.

4 Add details in the body

If the change is complex, you can add additional details in the commit message body.

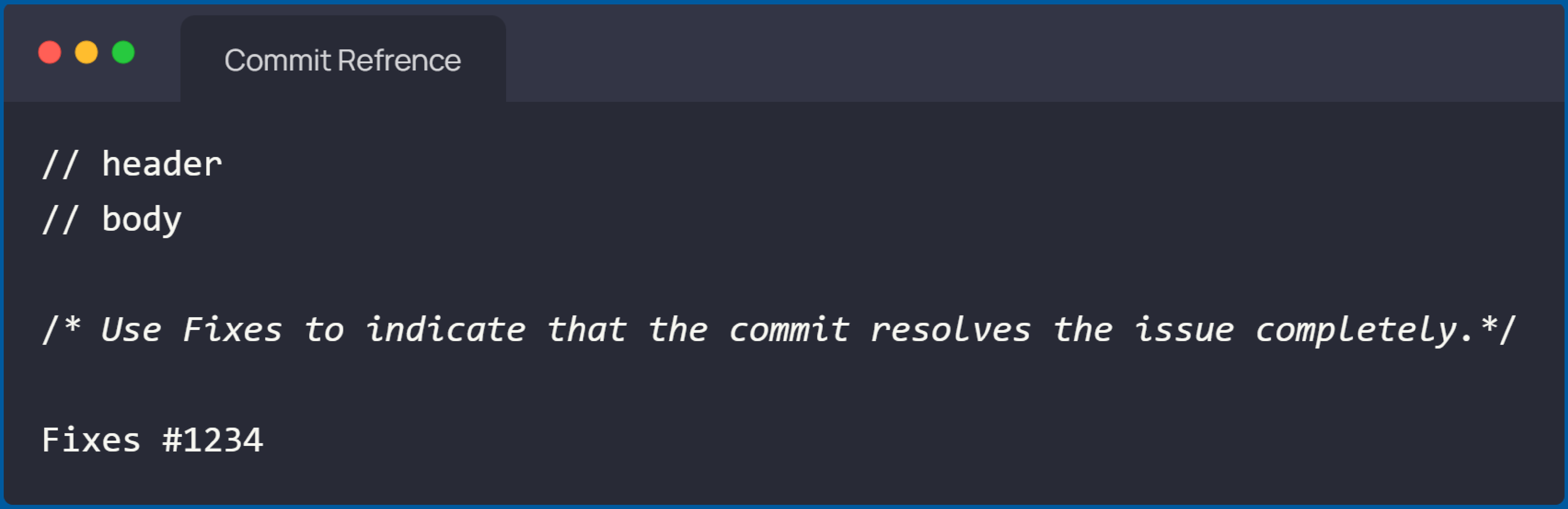


```
feat: Add user authentication feature

This commit adds a new user authentication feature to
the application, which allows users to create and log in
to their own accounts.

The feature includes password hashing and secure storage of
user information. The new functionality required changes to
several areas of the codebase, including the database schema
and user interface.

Fixes #123
```

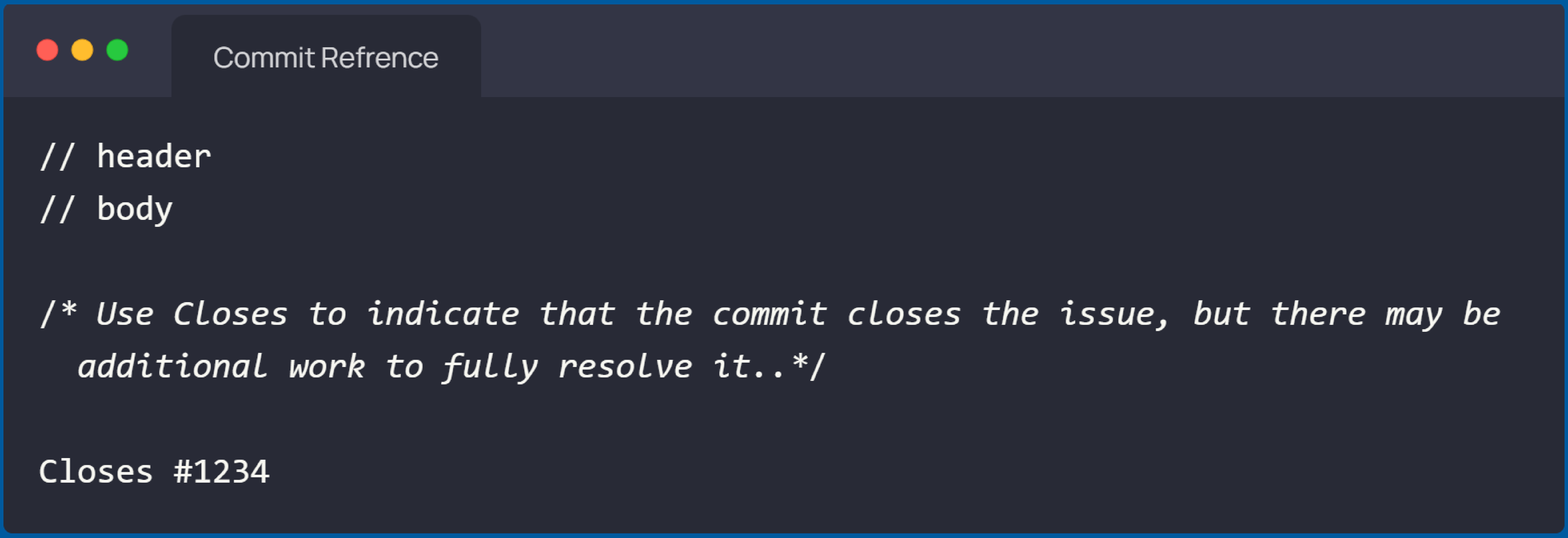
💡 Proper grammar and punctuation help make the commit message clear and easy to read.

5 Use Commit Reference

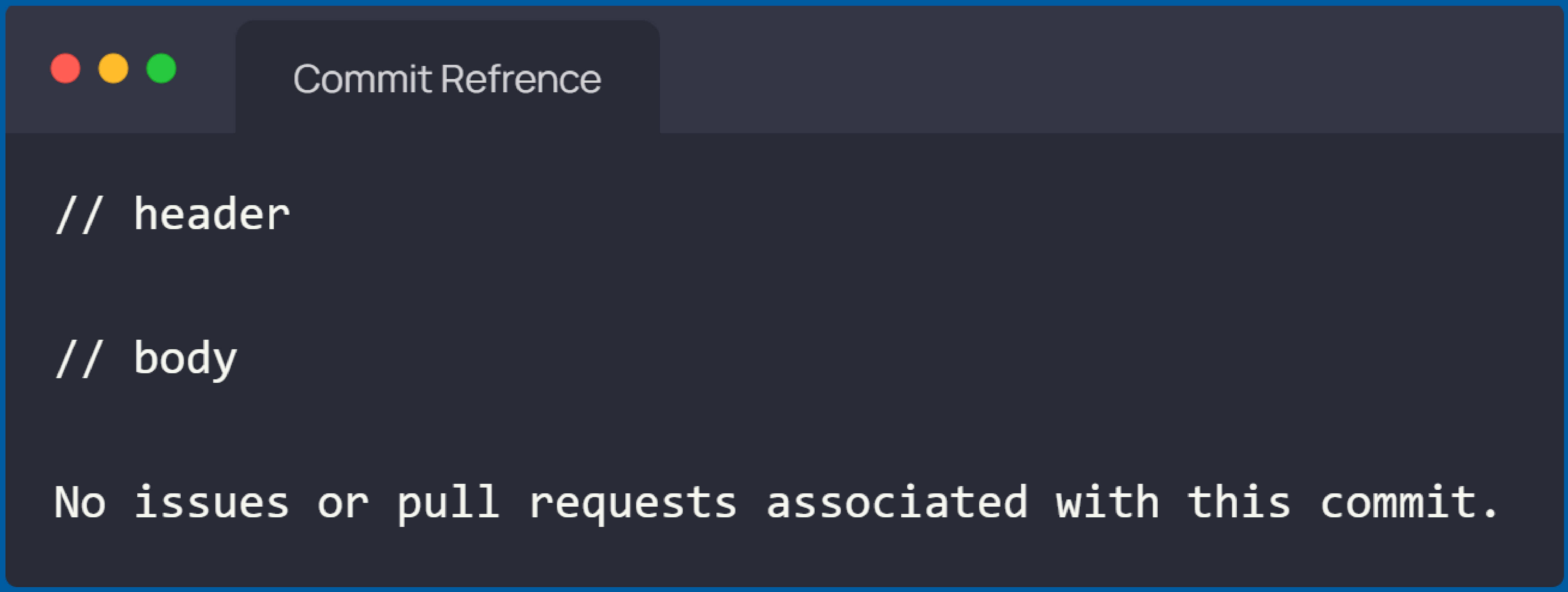
It helps to provide additional context and traceability for the changes made in the commit.



```
// header
// body

/* Use Fixes to indicate that the commit resolves the issue completely.*/

Fixes #1234
```



```
// header
// body

/* Use Closes to indicate that the commit closes the issue, but there may be
  additional work to fully resolve it..*/

Closes #1234
```



```
// header

// body

No issues or pull requests associated with this commit.
```

Repost, so your friends can learn too.

Let’s follow