> Reference: POE

You're absolutely right, my previous examples were not fully consistent with the best practice of using present tense and imperative mood for commit messages. Let me provide some updated examples that follow this guideline:

1. **New Feature**:

```
feat(authentication): implement user registration and login

- Add new user registration and login pages
- Integrate with the authentication API
- Implement session management to keep users logged in
```

2. **Bug Fix**:

```
fix(shopping-cart): correct issue with discount calculation

- Fix a bug where the discount is not being applied correctly
- Add unit tests to ensure the discount calculation works as expected
- Close #123
```

3. **Code Refactoring**:

```
refactor(api-client): migrate to new HTTP client library

- Replace the old HTTP client library with a more modern and robust solution
- Update all API client code to use the new library
- Improve error handling and response parsing
- Reduce code duplication and improve maintainability
```

4. **Documentation Update**:

```
docs: update README with installation and setup instructions

- Add steps for installing dependencies
- Include instructions for configuring the development environment
- Provide guidelines for running the application locally
```

5. **Chore (Maintenance Task)**:

```
chore: upgrade dependencies to latest versions

- Update all package dependencies to their latest stable versions
- Run automated tests to ensure compatibility with the new versions
- Make no functional changes, only dependency upgrades
```

By using present tense and imperative mood, the commit messages now read as instructions on what the commit will do, rather than what has been done. This helps maintain a consistent and clear style throughout the project's commit history.