



# Shrisht's Pull Request Guide (for EcomiqX)

This guide outlines the essential steps for creating and managing pull requests within the EcomiqX GitHub repository. Following these instructions ensures a smooth and efficient workflow for all contributors.

# Step 1: Push Your Feature Branch

Before initiating a pull request, ensure your local feature branch is pushed to the remote repository. This makes your changes visible to GitHub.

## Local Workflow

Shrisht's typical local Git commands:

```
git checkout develop
git pull origin develop
git checkout -b feature-login-ui
# Work on the code
git add .
git commit -m "Added login UI"
git push origin feature-login-ui
```

# Step 2: Open the Pull Request Page

There are two primary methods to open a pull request on GitHub once your feature branch is pushed.

1

## Method A: Using the Banner

Navigate to your repository in the browser:

<https://github.com/tecresearch/ecomiqx>

GitHub will display a banner: "feature-login-ui was recently pushed. Compare & pull request." Simply click on the "Compare & pull request" button.

2

## Method B: Manual Creation

Alternatively, go to "Pull Requests" in your GitHub repository and select "New Pull Request."

Set the **base branch** to develop and the **compare branch** to feature-login-ui. Then, click "Create Pull Request."

# Step 3: Fill in PR Details

Providing comprehensive and clear details in your pull request is crucial for effective review and collaboration.

## Title Format

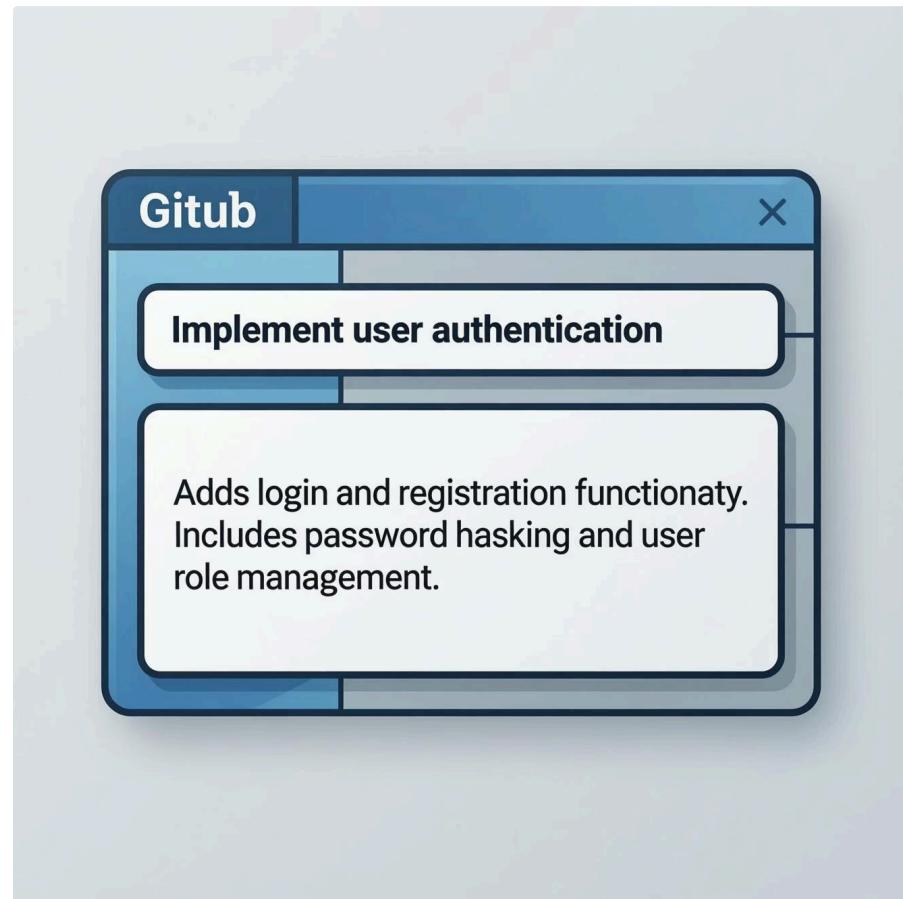
Adhere to a standardised title format for clarity:

```
[Feature] Login UI
```

## Description Template

Utilise the provided description template to detail your changes:

```
## What this PR does
- Added login UI page
- Integrated form with backend API
## Notes
- No base code changes
- Tested locally
```



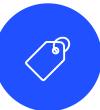
# Assign Reviewers and Labels

Proper assignment and labelling streamline the review process and categorise your pull request.



## Assign Reviewers

Always assign a reviewer to your pull request. For EcomiqX, Brijesh is the designated reviewer.

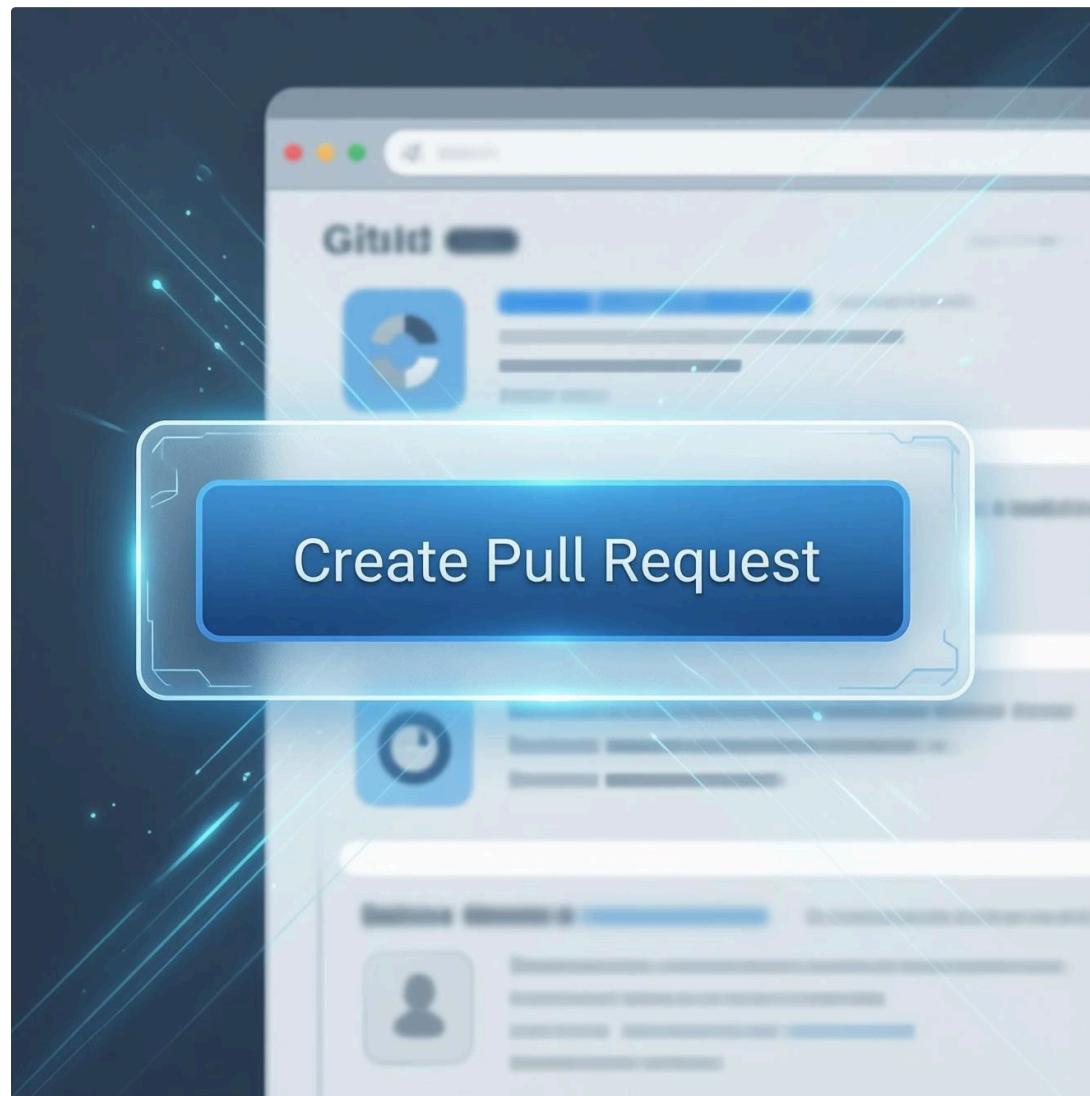


## Set Labels

Optionally, add relevant labels such as `feature` to help organise and filter pull requests.

# Step 4: Submit Your Pull Request

Once all details, reviewers, and labels are set, you can officially submit your pull request for review.



## Final Click

Click the "[Create Pull Request](#)" button. GitHub will then display the newly created PR page.

This action notifies the assigned reviewer and marks your changes ready for inspection.

# Step 5: Brijesh's Review Process

Brijesh, as the reviewer, undertakes a thorough examination of the submitted code to ensure quality and adherence to project standards.



## Code Inspection

Checks the overall quality, structure, and readability of the code.

## Base Code Integrity

Verifies that no unintended changes have been made to the base code.

## Clean Commits

Ensures commit history is clean, concise, and descriptive.

# Merging the Pull Request

Upon successful review and approval, the pull request will be merged into the develop branch.

## Approval

Once Brijesh approves the changes, the PR is ready for merging.

## Merge Action

The "Merge Pull Request" button becomes active, integrating the feature branch into develop.



# Step 6: Update Local Branch After Merge

After a pull request is merged, it's crucial for developers to synchronise their local develop branch with the remote repository.

01

## Checkout Develop

Switch to your local develop branch.

```
git checkout develop
```

02

## Pull Latest Changes

Fetch and integrate the latest changes from the remote develop branch.

```
git pull origin develop
```

03

## Synchronisation

Your local develop branch is now up-to-date with the main repository.

# Best Practices for EcomiqX PRs

Following these best practices will ensure a streamlined and effective contribution process within EcomiqX.



## Clear Communication

Always provide clear and concise explanations in your PR descriptions.



## Small, Focused PRs

Keep your pull requests small and focused on a single feature or bug fix.



## Self-Review

Perform a thorough self-review before requesting feedback from others.



## Collaborate

Engage actively with reviewers and address feedback promptly.

