

# Open Source contributions

Contributing to open-source can be a rewarding experience that improves your coding skills, helps you connect with the developer community, and builds your resume. Here's a **step-by-step** guide tailored for a beginner **front-end developer** with some GitHub expertise.

---

## Step 1: Understand Open Source Contribution

Before diving in, you should understand what open source means and how you can contribute.

### Ways to Contribute:

- Fixing bugs
  - Adding new features
  - Improving documentation
  - Writing tests
  - Helping with design/UI improvements
  - Reviewing PRs (Pull Requests)
  - Answering questions in discussions/issues
- 

## Step 2: Set Up Your GitHub Profile

Since you already have GitHub experience, ensure your profile is **presentable**:

✅ Add a **profile picture**

✅ Write a **bio** with skills & interests (e.g., "Frontend Developer | React, Vue, JavaScript")

✅ Pin a few **projects** you've worked on

---

## Step 3: Find the Right Open Source Project

Picking the right project will keep you motivated. Use these resources to discover beginner-friendly repositories:

## Where to Find Good First Issues

- ◆ [GitHub Explore](#) - Browse trending projects
- ◆ [First Contributions](#) - Great for beginners
- ◆ [Good First Issues](#) - Filters beginner-friendly issues
- ◆ [Awesome for Beginners](#) - A curated list of beginner-friendly projects

## How to Pick a Project

- Choose **frontend-related** projects (HTML, React, Tailwind, JavaScript, TypeScript)
- Check if the project is **active** (frequent commits, PR merges)
- Read the **README.md** file to understand the project
- Look for **good first issues** or **help wanted** labels

---

## Step 4: Set Up the Project Locally

Once you find a project you like:

### 1 Fork the repository

- Click the **Fork** button on GitHub (creates a copy in your GitHub account).

### 2 Clone the repository to your machine

```
git clone https://github.com/YOUR-USERNAME/PROJECT-NAME.git
```

### 3 Navigate into the project folder

```
cd PROJECT-NAME
```

### 4 Install dependencies (check readme first, or try command below)

```
npm install # or yarn install
```

### 5 Run the project

Check README file for instructions about how to run the project

### 6 Explore the codebase

- Read the **README.md** for setup instructions.
  - Look at the folder structure and understand the code.
- 

## Step 5: Pick an Issue & Start Coding

### Choosing an Issue

- Go to the project's **Issues** tab on GitHub.
- Filter by **"good first issue"** or **"help wanted"**.
- Read the issue carefully. If you have questions, ask the maintainers in the comments.

### Assign Yourself the Issue

- If the repo requires permission, **comment on the issue**: *"Hey! I'd love to work on this issue. Can you assign it to me?"*
  - Wait for a maintainer to assign it to you.
- 

## Step 6: Make Your Changes

### 1 Create a new branch

```
git checkout -b fix-bug-123
```

### 2 Make your changes

- If you're fixing a bug, modify the relevant files.
- If it's a UI change, test the layout.
- If it's documentation, edit `.md` files.

### 3 Run tests & check formatting

```
npm run test # If there are tests  
npm run lint # If the project has linting
```

## Step 7: Commit and Push Your Changes

### 1 Stage the changes

```
git add .
```

### 2 Commit with a meaningful message

```
git commit -m "Fix issue #123: Updated button color in navbar"
```

### 3 Push the changes to your fork

```
git push origin fix-bug-123
```

## Step 8: Create a Pull Request (PR)

### 1 Go to your forked repo on GitHub

### 2 Click "Compare & Pull Request"

### 3 Write a clear PR description

- What was the issue?
- What did you fix?
- How did you test it?

### 4 Submit the PR and wait for review 🎉

## Step 9: Respond to Feedback & Merge

- If maintainers request changes, update your code and push again:

```
git add .  
git commit --amend -m "Updated button color based on feedback"  
git push origin fix-bug-123 --force
```

- Once approved, your PR gets merged! 🏆

## Step 10: Keep Contributing & Engaging

- Find more issues and contribute regularly.

- Join the project's **Discord/Slack** if available.
- Star & share repositories you contribute to.

🎯 **Your first contribution is just the start!** Keep building your GitHub presence and learning new skills.