# Using GitHub Issues for Solo Backend Projects

## What Are GitHub Issues?

- GitHub Issues are tools for tracking bugs, planning features, and organizing tasks.
- Even in solo projects, they help you manage work professionally and build a visible history for recruiters.

## Why Use Issues (Even Solo)

- Document bugs and enhancements in one place.
- Plan your sprints and tasks like a real team would.
- Keep a professional and visible worklog.
- Use labels and checklists to organize your thinking.
- Create a project board to track status (To Do, In Progress, Done).

## How to Create an Issue

- Go to your repo > Issues tab > 'New Issue'.
- Give it a clear title: e.g., 'Bug: API crashes on empty title'.
- Write a structured description:
-   - Whats wrong or what should be built
-   - Steps to reproduce (for bugs)
-   - Expected result
-   - Optional checklist
- Use labels like `bug`, `enhancement`, or `priority: high` for clarity.

## Examples: Bug and Feature

- Bug Example:
-   Title: Bug: Contact form crashes on empty field
-   Description: Submitting the contact form with an empty email field crashes the app.
-   Expected: Error message should be shown to the user.
-
- Feature Example:
-   Title: Feature: Add user profile view
-   Description: Create a page that shows user info after login. Include profile picture and email address.
-   Checklist:
-     - [ ] Create profile.html

# Using GitHub Issues for Solo Backend Projects

- - [ ] Add GET route in Flask
- - [ ] Write unit test for /profile

## How to Integrate Issues in Your Workflow

- 1. Start your sprint by creating issues for each feature or task.

- 2. When starting a feature branch, link it to an issue (write issue # in your commit or PR).

- 3. Use issues to track bugs you discover while testing.

- 4. At sprint end, close completed issues when merged to main.

- 5. This gives a clear record of your process and priorities.