

GitHub Repository Structure for Bot Assignments

Repository Name: `ai-assistant-evolution`

This is your living portfolio of AI development. Every commit tells the story of your learning.

Required Directory Structure

```
ai-assistant-evolution/
|
├─ README.md           # Overview of your bot journey
├─ .gitignore          # Ignore system files
|
├─ assignment-1-bad-bot/ # H.2.1 Bad Bot
|   │
|   ├─ prompt-v1.md     # Your terrible bot prompt
|   ├─ test-outputs/    # Screenshots of disasters
|   │   │
|   │   ├─ disaster-1.png
|   │   ├─ disaster-2.png
|   │   └─ disaster-3.png
|   │
|   ├─ reflection.md    # What you learned from failure
|   └─ submission.md    # Link to all components
|
├─ assignment-2-good-bot/ # H.2.2 Good(ish) Bot
|   │
|   ├─ bad-bot-v1.md    # Original terrible prompt
|   ├─ good-bot-v2.md   # Improved prompt (ONE change highlighted)
|   ├─ test-cases.md    # Your three documented test scenarios
|   ├─ UX-IMPROVEMENT.md # Your improvement report
|   ├─ screenshots/    # Before/after evidence
|   │   │
|   │   ├─ test1-before.png
|   │   ├─ test1-after.png
|   │   ├─ test2-before.png
|   │   ├─ test2-after.png
|   │   ├─ test3-before.png
|   │   └─ test3-after.png
|   │
|   └─ submission.md    # Links everything together
|
└─ future-assignments/  # Space for growth
    └─ .gitkeep
```

GitHub Workflow Requirements

For EVERY Assignment

1. Create an Issue First

```
Title: "Assignment 2: Good Bot Implementation"
Body:
```

- [] Identify key problem
- [] Document test cases
- [] Make single change
- [] Test and screenshot
- [] Write reflection
- [] Submit PR

2. Create a Branch

```
Branch name: "2-good-bot"
Never work on main!
```

3. Commit Messages That Tell a Story

```
BAD: "updated stuff"
BAD: "fixed"
BAD: "asdfasdf"

GOOD: "Add initial test cases for OFFBYONE bot"
GOOD: "Implement footnote solution for number accuracy"
GOOD: "Add screenshots showing improvement in counting"
```

4. Pull Request for Submission

```
Title: "Assignment 2: OFFBYONE Improvement"
Body:
- Closes #[issue number]
- Summary of what you changed
- Link to key files
- Any notes for instructor
```

Main README.md Template

```
# AI Assistant Evolution Portfolio
*By: [Your Name]*

## About This Repository
This repository documents my journey learning AI development through iterative
improvement and intentional failure.

## Assignments

### 📄 Assignment 1: Bad Bot
**Bot Name:** [Your Bot Name]
**Completion Date:** [Date]
**Key Learning:** [One sentence about what you learned from failure]
[📄 View Assignment](./assignment-1-bad-bot/submission.md)

### 📄 Assignment 2: Good(ish) Bot
**Status:** In Progress
**Bot Name:** [Your Bot Name]
```

```
**Improvement Focus:** [What you're fixing]
[ View Assignment](./assignment-2-good-bot/submission.md)

### Future Assignments
- Assignment 3: [Coming Soon]
- Assignment 4: [Coming Soon]

## My AI Development Philosophy
[2-3 sentences about what you're learning about AI development]

## Best Failure So Far
[Screenshot or quote of your favorite disaster]

## Contact
- GitHub: [@yourusername]
- Course: CSC-113 AI Fundamentals
```

Submission Checklist (submission.md)

Each assignment folder needs a `submission.md` file:

```
# Assignment 2: Good(ish) Bot Submission

### Submission Checklist
- [ ] All required files present
- [ ] Screenshots properly named
- [ ] Test cases documented
- [ ] Changes clearly marked
- [ ] Reflection complete
- [ ] PR created and linked to issue

## Quick Links
- [Original Bot Prompt](./bad-bot-v1.md)
- [Improved Bot Prompt](./good-bot-v2.md)
- [Test Cases](./test-cases.md)
- [UX Improvement Report](./UX-IMPROVEMENT.md)
- [Screenshots Folder](./screenshots/)

## Summary
**Bot Name:** [Name]
**Single Change Made:** [One sentence]
**Success Rate:** [X/3 tests improved]
**Best Unintended Consequence:** [What broke in a funny way]

## Instructor Notes
[Any additional context or issues you encountered]
```

Git Commit History (What We're Looking For)

Good Commit Pattern:

2 hours ago - Add test case documentation for OFFBYONE bot
2 hours ago - Create initial bad bot prompt file
1 hour ago - Implement tiny footnote solution
1 hour ago - Add before screenshots for all test cases
45 min ago - Add after screenshots showing improvements
30 min ago - Write UX improvement analysis
15 min ago - Create submission file with all links
5 min ago - Final proofread and format fixes

Bad Commit Pattern:

5 min ago - Upload everything at once

Common Problems & Solutions

"I accidentally committed to main"

```
# Don't panic! Create new branch from current state
git checkout -b 2-good-bot
# Continue working
# Create PR from this branch to main
```

"I forgot to create an issue first"

Create it now, reference it in your PR. Better late than never.

"My screenshots are huge"

- Use PNG for text screenshots
- Crop to relevant parts
- Aim for < 500KB per image
- Name them descriptively

"I made multiple changes by accident"

- Pick the MOST impactful one
 - Document it as your single change
 - Mention others in "unintended consequences"
 - Or: Reset and do it right
-

Grading Visibility

Your GitHub repository shows:

- **Commit frequency** - Are you iterating or cramming?
- **Message quality** - Do you understand what you're doing?
- **Branch usage** - Following professional workflows?
- **Issue tracking** - Planning before implementing?
- **PR descriptions** - Can you explain your work?

All of these are part of your grade!

Pro Tips

1. **Commit After Each Section** Don't wait until everything is done. Commit test cases, then screenshots, then analysis.
2. **Use GitHub Issues as Your Todo List** Create checkboxes for each part of the assignment.
3. **Screenshot Everything** You think you'll remember that funny error. You won't.
4. **Write Commit Messages for Future You** "Fixed stuff" tells you nothing in 2 weeks.
5. **The Repository IS Your Portfolio** Potential employers will look at this. Make it professional but personality-filled.

The Meta-Learning

This structure teaches you:

- Version control as documentation
- Professional development workflows
- Incremental progress tracking
- Clear communication through commits
- Portfolio building while learning

Every commit is evidence of learning. Every branch shows experimentation. Every PR demonstrates completion.

Remember: In the real world, your GitHub history IS your resume.

"Your repository should tell the story of your learning - failures, fixes, and all."