Quick Start Guide

✅ What's Been Done

1. **3 Milestones Created** in your GitHub repository:
   - Phase 1: Foundation
   - Phase 2: Database & Features
   - Phase 3: Deployment & Production

2. **9 Comprehensive Issues** generated as markdown files
   - Each issue is 300-900+ lines
   - Complete with code examples, steps, and acceptance criteria
   - AI-agent-friendly format

3. **Helper Scripts** created:
   - `create_issues.sh` - Automated issue creation
   - `README.md` - Full documentation
   - `SUMMARY.md` - Project overview

🚀 What to Do Now

Step 1: Review the Issues (Optional)

Browse the issue files to see what's included:

```
cd /home/ubuntu/pp-bot-issues
ls -la issue-*.md
```

Open any file to see the detailed content:

```
cat issue-01.md  # Slack App Setup Guide
cat issue-02.md  # TypeScript Migration
# ... etc
```

Step 2: Create the Issues in GitHub

Option A: Using GitHub CLI (Fastest) ⭐

```
# Make sure you're authenticated
gh auth login

# Run the creation script
cd /home/ubuntu/pp-bot-issues
./create_issues.sh
```

This will automatically create all 9 issues in your repository with the correct milestones and labels.

Option B: Manual Creation (If you prefer)

1. Go to https://github.com/stevencarpenter/pp-bot/issues/new

2. Open `issue-01.md` in this directory
3. Copy the title and body
4. Paste into GitHub
5. Set milestone to “Phase 1: Foundation”
6. Add labels: documentation, setup, good-first-issue
7. Submit
8. Repeat for issues 02-09

Step 3: Start Working on Issues

Recommended order:

Week 1: Foundation
1. Issue #1 - Slack App Setup Guide (2-3 hours)
2. Issue #2 - Migrate to TypeScript (4-6 hours)
3. Issue #7 - Error Handling & Logging (3-4 hours)

Week 2: Database & Features
4. Issue #3 - PostgreSQL Integration (6-8 hours)
5. Issue #8 - Testing Infrastructure (5-6 hours)
6. Issue #4 - Leaderboard Command (3-4 hours)

Week 3: Deployment
7. Issue #5 - Railway.com Deployment (2-3 hours)
8. Issue #6 - CI/CD with GitHub Actions (3-4 hours)
9. Issue #9 - Documentation (4-5 hours)

📋 Verification Checklist

After creating issues in GitHub, verify:

- [ ] All 9 issues appear at https://github.com/stevencarpenter/pp-bot/issues
- [ ] Milestones are correctly assigned
- [ ] Labels are applied
- [ ] Issue descriptions are properly formatted

📚 Documentation

- **README.md** - Complete guide on how to use the issues
- **SUMMARY.md** - Overview of what was created
- **QUICK_START.md** - This file

🤖 Using with AI Agents

Each issue can be assigned to an AI agent (like Cursor, GitHub Copilot, or Abacus.AI DeepAgent) for autonomous implementation. The issues contain everything needed:

- Complete context
- Step-by-step instructions
- Code examples
- Acceptance criteria

- Testing guidance

💡 Tips

1. **Start with Issue #1** - It’s marked as `good-first-issue` and sets up the foundation
2. **Follow the order** - Issues have dependencies on each other
3. **Test as you go** - Don’t skip the testing sections
4. **Review PRs carefully** - Especially for database and deployment changes
5. **Ask questions** - Each issue has references to documentation

📞 Need Help?

If you encounter issues:

1. Check the README.md for detailed instructions
2. Verify GitHub CLI is installed: `gh --version`
3. Ensure you’re authenticated: `gh auth status`
4. Check repository access: `gh repo view stevencarpenter/pp-bot`

🎉 What You’ll Build

By completing all 9 issues, you’ll transform pp-bot into:

- ✅ Production-ready TypeScript application
- ✅ PostgreSQL-backed persistent storage
- ✅ Deployed on Railway.com
- ✅ Automatic CI/CD with GitHub Actions
- ✅ Comprehensive testing (80%+ coverage)
- ✅ Professional error handling and logging
- ✅ Complete documentation

📈 Estimated Timeline

- **1 week (40 hours):** Full implementation by experienced developer
- **2 weeks (part-time):** 20 hours per week
- **3 weeks (casual):** 10-15 hours per week

Next Command

```
cd /home/ubuntu/pp-bot-issues
./create_issues.sh
```

Or if you prefer manual creation, start with:

```
cat issue-01.md
```

Good luck! 🚀