






Week 2: Start Guide

Date: January 26, 2025

Status:  Ready to Start


Total Time: 7 hours

Current Status

-  **Week 1 Complete** - Accessibility fixes and AI Assistant implemented
-  **Documentation Complete** - All plan documents ready
-  **Cleanup Complete** - Obsolete files removed
-  **Week 2 Ready** - Plan documented and ready to execute

Week 2 Overview

Goal: Integration testing, performance verification, and documentation updates

Risk Level:  Low (mostly testing and documentation, minimal code changes)

Tasks Breakdown

- 2.1: Manual Testing & Validation** (3 hours)
- 2.2: Performance Optimization** (2 hours)
- 2.3: Documentation Review & Updates** (2 hours)

Immediate Next Steps

Step 1: Read Preparation Documents (10-15 min)

- Read:** docs/README-8-WEEK-PLAN.md
 - Understand the workflow
 - Review testing checklist
- Review:** docs/CURSOR-PROTECTION-GUIDE.md
 - Know which files NOT to modify
 - Understand testing procedures
- Open:** docs/8_WEEK_PLAN_CORRECTED.md
 - Go to Week 2 section
 - Review all 3 tasks

Step 2: Create Git Branch (1 min)

```
git checkout -b week-2-testing-refinement
```

Step 3: Start Task 2.1 - Manual Testing & Validation (3 hours)

Goal: Verify all systems work as expected

Testing Checklist:

Desktop Testing

- ☐ Chrome → Chat widget visible and functional
- ☐ Firefox → Chat widget visible and functional

- ☐ Safari → Chat widget visible and functional
- ☐ Click button → Opens correctly
- ☐ Type message → Sends correctly
- ☐ Claude responds → Real-time streaming works
- ☐ History persists → Refresh test (conversation saved)

Mobile Testing

- ☐ iPhone → Widget responsive and works
- ☐ Android → Widget responsive and works
- ☐ Click button → Works correctly
- ☐ Type message → Works correctly
- ☐ Responses → Stream properly

Error Testing

- ☐ Network disconnected → Error message appears
- ☐ Rate limit hit → "Rate limit" message appears
- ☐ Invalid input → Handled gracefully

Integration Testing

- ☐ Widget on homepage → Visible and works
- ☐ Widget on all pages → Visible and works
- ☐ Conversations stored in Supabase → Verify in dashboard
- ☐ No console errors → F12 → Console clean
- ☐ No TypeScript errors → `npm run build` passes

Files to Test (DO NOT MODIFY):

- `app/api/ai-assistant/route.ts`
- `components/AIAssistantWidget.tsx`
- `app/layout.tsx`

Output: Testing report documenting all verified functionality

Step 4: Task 2.2 - Performance Optimization (2 hours)

Goal: Ensure chat widget doesn't impact page load performance

Performance Checks:

1. Run Lighthouse Before Optimization:

Open browser: F12 → Lighthouse → Generate Report

- ☐ Record current scores (Accessibility, Performance, etc.)
- ☐ Target: Maintain 91-98/100

2. Load Time Analysis:

- ☐ First Contentful Paint (FCP)
- ☐ Largest Contentful Paint (LCP)
- ☐ Cumulative Layout Shift (CLS)
- ☐ Time to Interactive (TTI)

3. Bundle Analysis:

- ☐ Check chat widget bundle size
- ☐ Check Claude library impact
- ☐ Check Supabase impact
- ☐ Optimize if needed

Files to Analyze (READ ONLY):

- `app/layout.tsx` (widget loading)
- `components/AIAssistantWidget.tsx` (component optimization)

Modifications Allowed (if needed):

- Add `React.memo()` to `AIAssistantWidget` if needed
- Lazy load widget if necessary
- Optimize imports (only if impacts performance)

DO NOT CHANGE:

- Model (claude-3-haiku-20240307)
- API endpoint
- Supabase integration
- Error handling

Output: Performance report with Lighthouse scores (before/after)

Step 5: Task 2.3 - Documentation Review & Updates (2 hours)

Goal: Ensure documentation matches current implementation

Files to Review:

- ☐ `docs/AI_ASSISTANT_WIDGET_DEPLOYMENT.md` (if exists)
- ☐ `docs/AI_ASSISTANT_TESTING_GUIDE.md` (if exists)
- ☐ `docs/LIGHTHOUSE_AUDIT_GUIDE.md`
- ☐ `docs/CASE_STUDY_PDF_GUIDE.md`
- ☐ `README.md` (root level)

Documentation Updates Needed:

- ☐ Update deployment guide with Haiku model info
- ☐ Document Claude 3 Haiku (current model)
- ☐ Note: Sonnet requires plan upgrade
- ☐ Update troubleshooting for model access (402, 403, 429 errors)
- ☐ Document credit verification steps
- ☐ Add links to Anthropic console

Create New Documentation:

- ☐ `docs/DEPLOYMENT_CHECKLIST.md` (what to verify before launch)
- ☐ `docs/MODEL_SELECTION_GUIDE.md` (Haiku vs Sonnet decision guide)
- ☐ `docs/ARCHITECTURE_OVERVIEW.md` (system diagram)

Quick Fix:

- ☐ Update comment in `app/api/ai-assistant/route.ts` line 6:
 - Change: "Claude 3.5 Sonnet API"
 - To: "Claude 3 Haiku API"

Output: Updated documentation ready for sharing

✓ Before Each Commit

Testing Checklist (from CURSOR-PROTECTION-GUIDE.md):

- ☐ `npm run build` (TypeScript check - no errors)
 - ☐ `npm run dev` (local test - server starts)
 - ☐ Manual feature test (test what you changed)
 - ☐ F12 console check (no red errors)
 - ☐ Lighthouse check (if modified layout/styles - maintain >91)
 - ☐ `git diff` (review changes)
 - ☐ Clear commit message: `feat(week-2): description`
 - ☐ `git push`
-

Expected Deliverables

By end of Week 2, you should have:

1. ✓ **Testing Report** - Documented all verified functionality
 2. ✓ **Performance Report** - Lighthouse scores, bundle analysis
 3. ✓ **Updated Documentation** - All docs match current implementation
 4. ✓ **New Documentation** - Deployment checklist, model guide, architecture overview
 5. ✓ **API Comment Fix** - Updated route.ts comment
-

Success Criteria

Week 2 is successful when:

- ✓ All testing complete (desktop, mobile, errors, integration)
 - ✓ Performance verified (Lighthouse maintained or improved)
 - ✓ Documentation updated (matches implementation)
 - ✓ 0 regressions introduced (existing features still work)
 - ✓ Ready for Week 3 (case study development)
-

Quick Help

If something breaks during testing: → Check `docs/CURSOR-PROTECTION-GUIDE.md` → "Emergency Recovery" section

If you need to modify a file: → Check `docs/CURSOR-PROTECTION-GUIDE.md` first to see if it's protected

If you're not sure about a task: → Reference `docs/8_WEEK_PLAN_CORRECTED.md` → Week 2 section for details

Ready to Start?

Current Status: ✓ All systems ready

Confidence: 95% (plan is corrected and aligned with codebase)

Next Action: Start with Task 2.1 - Manual Testing & Validation

Remember:

- Week 2 is mostly testing and documentation (low risk)
- No major code changes required
- Focus on verification and documentation

- Test everything before committing

YOU'VE GOT THIS. 🙌

Start Date: [Your start date]

Target Completion: [Start date + 7 hours]

Status: Ready to begin 🚀