# Build a Broken Link Checker with Cursor AI

Create a free, no-code broken link checker tool using Cursor AI.

Based on Build A Free Broken Link Checker Tool With Cursor Ai No Coding Experience Needed by Casey Builds It

---

## 1. Understanding the Need for a Broken Link Checker

*Recognize the impact of broken links and the benefits of a custom tool.*

- ■ Acknowledge that broken links negatively affect SEO.
- ■ Understand that broken links harm user experience.
- ■ Identify the cost and learning curve associated with paid crawlers.
- ■ Consider building a lightweight alternative to paid tools.
- ■ Recognize the value of a custom, free SEO toolkit.

## 2. Tool Capabilities and Features

*Understand what the broken link checker tool does.*

- ■ Scan a website for broken links.
- ■ Identify both internal broken links.
- ■ Identify external broken links.
- ■ Utilize a simple Graphical User Interface (GUI).
- ■ Export scan results automatically to a CSV file.
- ■ Name CSV files with domain and timestamp for organization.
- ■ Convert the Python script to a standalone EXE.

## 3. Getting Started with Cursor AI

*Prepare to use Cursor AI to generate the tool's script.*

- ■ Access Cursor AI for tool generation.
- ■ Understand the prompt required to generate the Python script.
- ■ Follow the walkthrough to prompt Cursor AI.
- ■ Ensure the generated script includes GUI functionality.
- ■ Verify the export functionality is included in the script.

## 4. Building the Tool's Interface

*Focus on creating the user-friendly GUI.*

- ■ Prompt Cursor AI to build the GUI.
- ■ Ensure the GUI is simple and easy to use.
- ■ Confirm the GUI allows for running scans.
- ■ Check that scan results are displayed in the GUI.

## 5. Implementing Export Functionality

*Set up the automatic export of scan results.*

- Configure the tool to automatically export results.
- Ensure results are exported to a CSV file.
- Verify the CSV file is named with the website's domain.
- Confirm the CSV file includes a timestamp.
- Organize scanned results for multiple site checks.

## 6. Testing the Broken Link Checker

*Run the tool to verify its functionality.*

- Execute a live scan on a real website.
- Observe the tool identifying actual broken links.
- Review the generated CSV report for accuracy.
- Test the GUI for ease of use during the scan.
- Confirm the standalone EXE runs without Python installation.

## 7. Enhancing the Tool (Future Steps)

*Consider potential improvements and expansions for the tool.*

- Explore scheduling automated scans.
- Investigate using a JSON config file for multiple sites.
- Consider integrating this tool into a larger SEO toolkit.
- Plan for future builds in the free SEO toolkit series.

## 8. Finalizing the Tool

*Prepare the tool for practical use.*

- Convert the Python script into a standalone EXE.
- Test the EXE on a Windows machine without Python.
- Hand off the tool to a client or teammate if applicable.
- Keep the tool as part of your custom SEO toolkit.

## Resources & Further Reading

*Use these pages while working through the checklist:*

- Casey Builds It

https://caseybuildsit.com/the-files/

- Casey Builds It

https://caseybuildsit.com/make-contact/

- Casey Builds It

https://caseybuildsit.com/

- Downloads

https://caseybuildsit.com/downloads/

---

Casey Builds It | -
- | caseybuildsit.com
Build your own free SEO tools with AI, starting with a broken link checker.