

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

Learn why checking for broken links is crucial for SEO and user experience.

- Acknowledge that broken links negatively impact SEO.
- Recognize that broken links harm user experience.
- Understand that paid crawlers are not always necessary.
- Identify the goal: build a lightweight broken link checker.
- Consider the benefits for SEO professionals and freelancers.
- Explore the advantage of building custom SEO toolkits.

2. Tool Functionality Overview

Understand what the broken link checker tool will do.

- Scan a target website for broken links.
- Identify both internal broken links.
- Identify external broken links.
- Provide a graphical user interface (GUI) for ease of use.
- Export scan results automatically to a CSV file.
- Name CSV files with domain and timestamp for organization.

3. Prompting Cursor AI for the Script

Learn how to use Cursor AI to generate the core Python script.

- Access and use the Cursor AI tool.
- Formulate prompts to generate the Python script for the link checker.
- Ensure prompts cover the core scanning logic.
- Review the generated Python script for accuracy.

4. Building the Graphical User Interface (GUI)

Follow the steps to add a user-friendly interface to the tool.

- Prompt Cursor AI to build the GUI.
- Integrate the GUI with the link checking functionality.
- Ensure the GUI is simple and intuitive.
- Test the GUI for proper display and interaction.

5. Implementing CSV Export Functionality

Set up the tool to export scan results.

- Prompt Cursor AI to add CSV export.
- Configure the export to create a clean CSV file.
- Ensure the filename includes the website domain.
- Implement timestamping for the CSV filename.
- Verify that results are correctly written to the CSV.

6. Testing the Broken Link Checker

Run the tool to ensure it functions as expected.

- Execute the tool on a real website.
- Observe the scanning process.
- Verify that broken internal links are flagged.
- Verify that broken external links are flagged.
- Check the generated CSV file for accurate results.

7. Creating a Standalone Executable

Convert the Python script into a standalone application.

- Follow instructions to convert the Python script.
- Create a standalone EXE file.
- Confirm the EXE runs without requiring Python installation.
- Test the standalone EXE on a Windows machine.
- Ensure the EXE maintains all original functionality.

8. Future Enhancements and Considerations

Explore potential improvements and next steps for the tool.

- Consider scheduling automated scans.
- Investigate using a JSON config file for multiple sites.
- Think about integrating this tool into a larger SEO toolkit.
- Evaluate adding more advanced checking features.
- Plan for future iterations based on user needs.

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 custom SEO tools efficiently with AI.