

Code Assistant ? Single?File Demo Documentation

Table of Contents

1. [Overview](#overview)
2. [Form Structure](#form-structure)
3. [Submission Flow](#submission-flow)
4. [Response Rendering](#response-rendering)
5. [Pros & Cons](#pros--cons)
6. [Overall Review](#overall-review)

- - -

1. Overview

The **Code Assistant** demo is a lightweight web interface that lets users upload a single source?code file, describe what they want the AI to do with it (e.g., explain, refactor, test), optionally specify a tech stack, and receive an AI?generated analysis. It is intended as a quick showcase of Recall AI?s code?analysis capabilities.

- - -

2. Form Structure

Element	Purpose	Notes
----- ----- -----		
`<h2>3. Code Assistant (single-file demo)</h2>`	Section title	Indicates the third feature in a series.
`<form action="/code-analyze" method="post" enctype="multipart/form-data">`	Submission endpoint	Uses `POST` with multipart encoding to handle file uploads.
`<input type="file" id="code_file" name="code_file" required>`	File picker	Requires a file; only one file is accepted.
`<textarea id="code_goal" name="code_goal">`	Goal description	Optional placeholder guides the user.
`<input type="text" id="tech_stack" name="tech_stack">`	Tech stack hint	Optional; can help the AI tailor its response.
`<button type="submit" class="primary">Analyze Code</button>`	Submit button	Triggers the form submission.

The form is wrapped in a `<div class="card">` for visual styling, giving it a distinct UI card look.

- - -

3. Submission Flow

1. **User Interaction**

- Upload a code file (*.py, *.js, etc.).
- Enter a goal (e.g., ?Explain what this code does and suggest improvements.?).
- Optionally specify a tech stack (e.g., ?React + Node.js + MongoDB?).
- Click **Analyze Code**.

2. **Server Handling (`/code-analyze`)**

- Receives the multipart form data.
- Extracts `code_file`, `code_goal`, and `tech_stack` .
- Sends the file content + prompt to Recall AI's code?analysis endpoint.
- Awaits the AI response.

3. **Response Delivery**

- Renders a new page (or updates the same page) with the AI's answer.
- Uses `{% if code_answer %}` to conditionally display results.

- - -

4. Response Rendering

When `code_answer` is present, the following card appears:

```
```html
<div class="card">
 <h2>Code Assistant Answer</h2>
 <p>Goal: {{ code_goal }}</p>
 <pre>{{ code_answer }}</pre>
</div>
```

```

- **Goal Echo** ? Re?shows the user?s goal for context.

- **P
snipp

- - -

5. Pros & Cons

Pros

- **Simplicity** ? One?page, minimal UI; easy to embed in demos or documentation.

- **In

Cons

- **Single?File Limitation** ? Cannot analyze multi?file projects or dependencies.

- **N
UX.

--

6. Overall Review

The **Code Assistant (single?file demo)** is a clean, functional showcase of Recall AI?s code analysis. Its minimalistic design makes it ideal for quick demonstrations or embedding in larger documentation sites. However, the demo would benefit from a few UX improvements:

1. **Add a loading indicator** to signal that the AI is processing.
2. **Validate file types** and provide user?friendly error messages.
3. **Support multi?file uploads** or a ZIP archive for more realistic projects.
4. **Implement AJAX submission** to avoid full page reloads and improve responsiveness.

Despite these areas for enhancement, the demo effectively demonstrates how a user can interact with an AI code assistant: upload a file, specify intent, and receive actionable insights. It serves as a solid foundation for building more advanced code?analysis tools.