Docky

GPT Code Snippets for Documentation

George Botros · Christian Ishimwe · Matthew Yin · Michael Imevbore

CIS 3500 Final Project

Problem & Solution

Problem

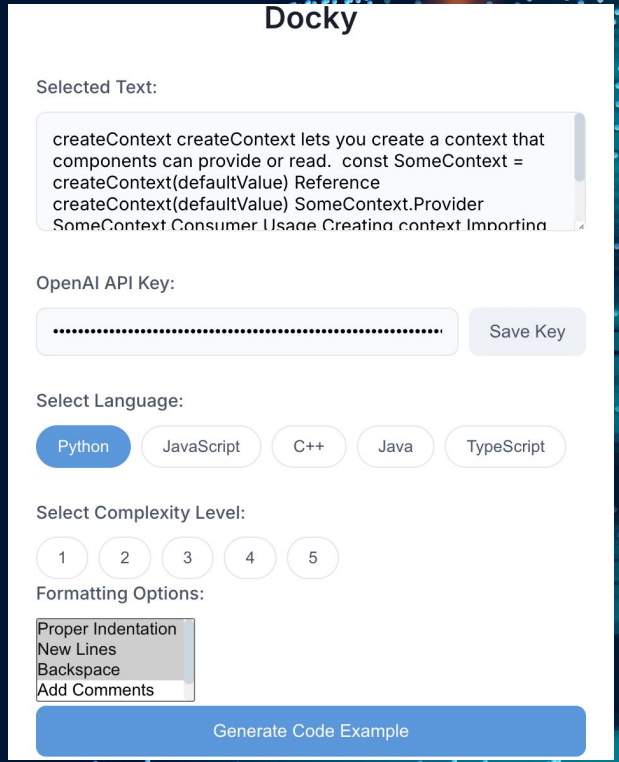
- Documentation sites often lack tailored, contextual code examples.
- Developers waste time hunting for sample code.
- Learners and cross-language devs struggle to bridge conceptual gaps.

Solution

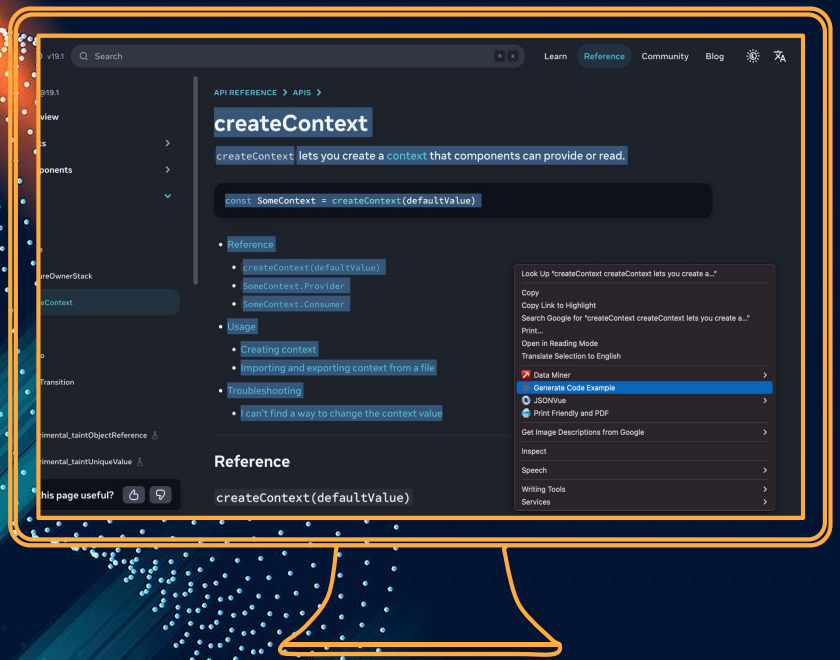
- Docky Chrome extension
- On-page AI-powered snippet generator
- Users select context and language

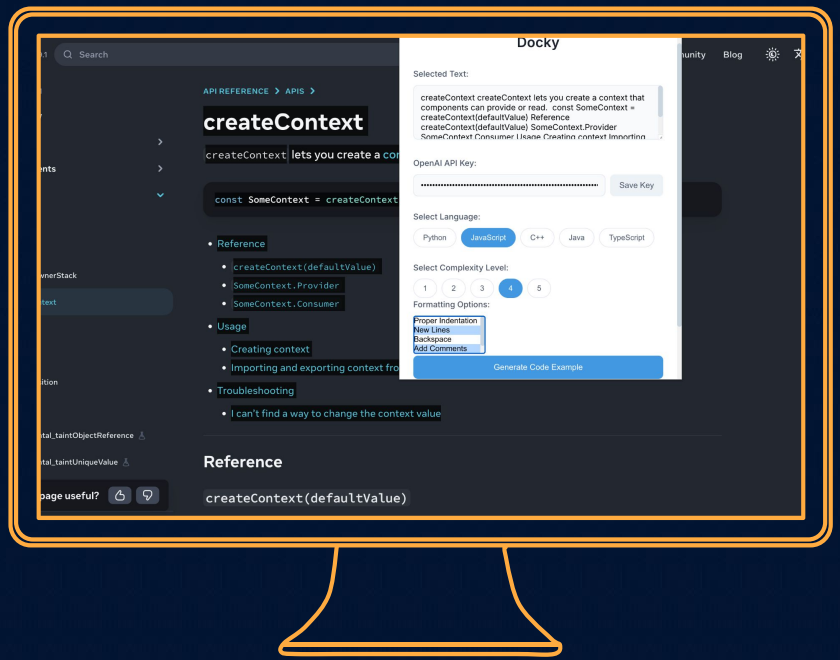
Key Features

- **Context-Aware Generation**
  - Extracts the exact docs text you're ready
- **Language & Complexity Control**
  - Choose any language and difficulty
- **Formatting Options**
  - Indentation, blank lines, explanatory comments on demand
- **Non-Intrusive UI**
  - Popup UI when you use the extension that shows you the generated snippet

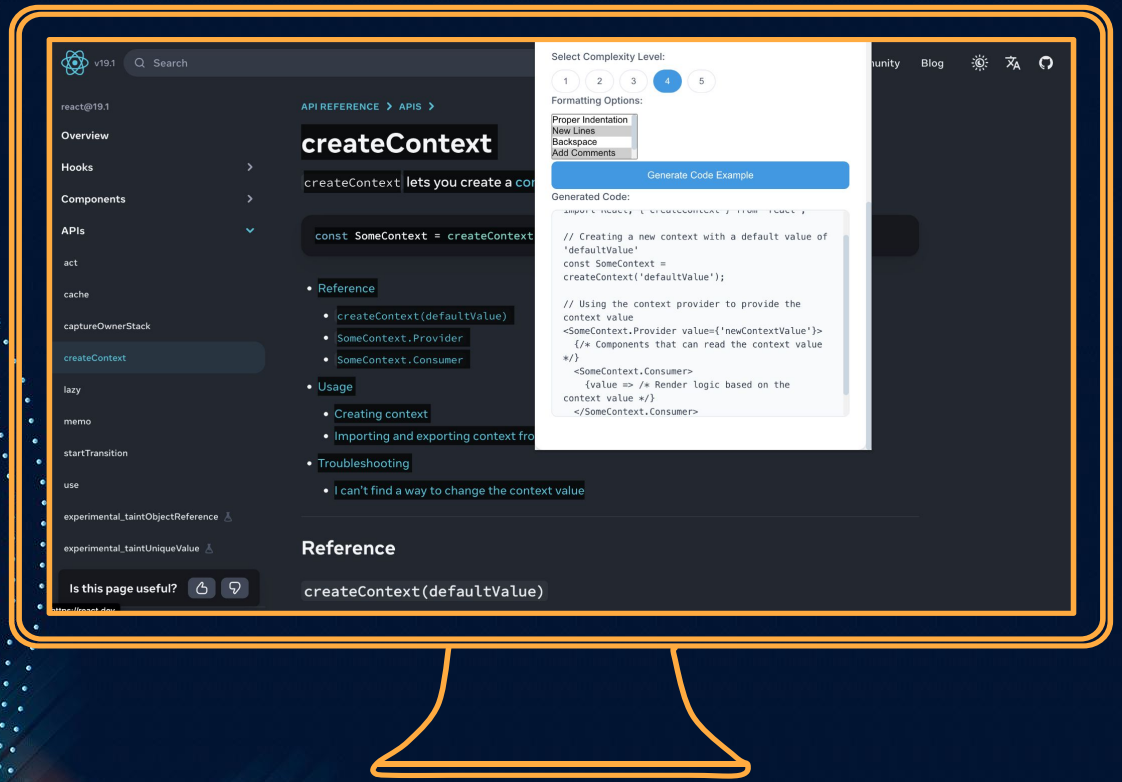


SCREENSHOTS





SCREENSHOTS

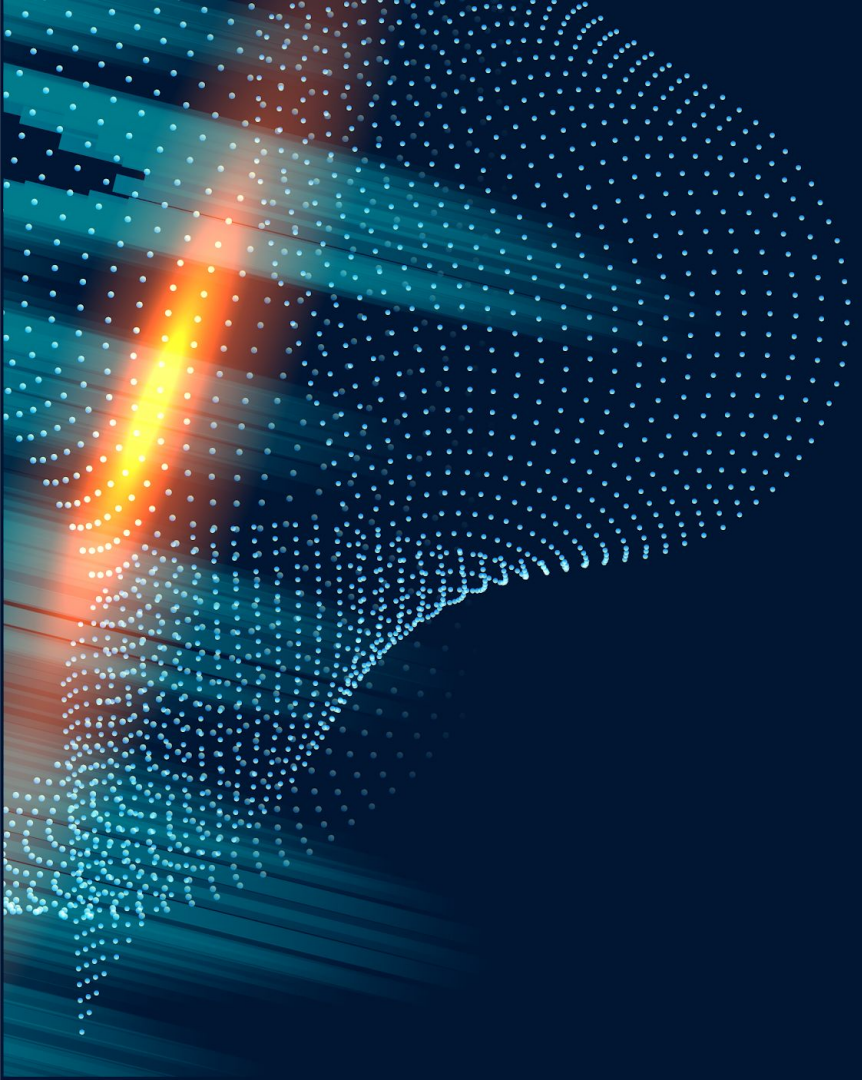


Architecture

- **Popup (UI Module)**
  - popup.html + popup.js
- **Reads/saves API key**
  - popup.js
- **AI Service (aiService.js)**
  - generateCodeSnippet() → wraps OpenAI Chat API call
  - cleanupCodeSnippet() → strips fences, trims
- **Background + Storage**
  - background.js for context menu
  - chrome.storage.local stores API key

Challenges & Technical Decisions

- **Trouble with Module Loading in the Extension**
  - Chrome pop-up scripts default to classic <script>—we needed full ES-module support
  - Added type="module" to popup.html and verified correct relative paths for all imports
- **Robust Prompt Engineering**
  - Defined a clear system prompt for consistent coding style
  - Layered user prompts to include language, complexity, and optional formatting directives
  - Iterated on prompt wording to minimize “hallucinations” and maximize snippet relevance
- **Comprehensive Test Environment**
  - Autoloaded .env variables in Jest via a dedicated jest.setup.js so we can use API keys



Thanks!