```
zipscribe_dfh3vcpr/
  EchoForge/
    .env.local
    .gitignore
    App.tsx
    README.md
    ROADMAP.md
    UPGRADEME.md
    constants.ts
    file_links.json
    generate_file_links.cjs
    index.html
    index.tsx
    metadata.json
    package-lock.json
    package.json
    tsconfig.json
    vite.config.ts
    components/
      AIActionMenu.tsx
      AnalyzeButton.tsx
      CodeDisplay.tsx
      DependencyViewer.tsx
      DiffViewer.tsx
      ExecutionConsole.tsx
      ExtensionGuideModal.tsx
      FileGraph.css
      FileGraph.tsx
      GeneratedTestViewer.tsx
      Header.tsx
      Icon.tsx
      MarkdownViewer.css
      MarkdownViewer.tsx
      RevisionHistoryViewer.tsx
      SettingsModal.tsx
      Sidebar.css
      Sidebar.tsx
      SuggestionsDisplay.tsx
    constants/
      themes.ts
    contexts/
      SettingsContext.tsx
      ThemeContext.tsx
    public/
      execution.js
      linter.js
    services/
      geminiService.ts
```

App.tsx

```
import React, { useState, useCallback, useContext, useEffect, useRef } from 'react';
import { analyzeCode, analyzeSelection, AIAction, AIGeneratedCode, analyzeDependencies, generateRequirementsFile, Depen
import { PYTHON_CODE } from './constants';
import Header from './components/Header';
import CodeDisplay from './components/CodeDisplay';
import SuggestionsDisplay from './components/SuggestionsDisplay';
import AnalyzeButton from './components/AnalyzeButton';
import { SettingsProvider, SettingsContext, Settings } from './contexts/SettingsContext';
import { ThemeProvider, ThemeContext } from './contexts/ThemeContext';
import ExecutionConsole from './components/ExecutionConsole';
import { FileGraph } from "./components/FileGraph";
import { Sidebar } from "./components/Sidebar";
import { MarkdownViewer } from "./components/MarkdownViewer";

type ExecutionMessage = {
    type: 'stdout' | 'stderr' | 'error' | 'done';
    data: string;
}
```

```
export type Revision = {
    id: string;
    code: string;
    timestamp: number;
    summary: string;
};


const AppContent: React.FC = () => {
  const [suggestions, setSuggestions] = useState<string | AIGeneratedCode | null>(null);
  const [originalSelection, setOriginalSelection] = useState<string>('');
  const [isLoading, setIsLoading] = useState<boolean>(false);
  const [error, setError] = useState<string | null>(null);
  const [code, setCode] = useState<string>(PYTHON_CODE);
  const [lastAction, setLastAction] = useState<AIAction | null>(null);

  const [dependencies, setDependencies] = useState<Dependency[] | null>(null);
  const [isAnalyzingDependencies, setIsAnalyzingDependencies] = useState<boolean>(false);
  const [requirementsFile, setRequirementsFile] = useState<string | null>(null);

  const [isExecuting, setIsExecuting] = useState<boolean>(false);
  const [executionOutput, setExecutionOutput] = useState<{type: 'stdout' | 'stderr' | 'error', message: string}[]>([]);
  const executionWorkerRef = useRef<Worker | null>(null);

  const [revisionHistory, setRevisionHistory] = useState<Revision[]>([]);
  const [breakpoints, setBreakpoints] = useState<Set<number>>(new Set());

  const { settings } = useContext(SettingsContext);
  const { theme } = useContext(ThemeContext);

  useEffect(() => {
    try {
        const savedHistory = localStorage.getItem('ai_refactor_history');
        if (savedHistory) {
            setRevisionHistory(JSON.parse(savedHistory));
        }
    } catch (e) {
        console.error("Failed to load revision history from localStorage", e);
        setRevisionHistory([]);
    }
  }, []);

  const addRevision = useCallback((summary: string, newCode: string) => {
    const newRevision: Revision = {
        id: crypto.randomUUID(),
        code: newCode,
        summary,
        timestamp: Date.now(),
    };
    setRevisionHistory(prev => {
        const updatedHistory = [newRevision, ...prev];
        try {
            localStorage.setItem('ai_refactor_history', JSON.stringify(updatedHistory));
        } catch (e) {
            console.error("Failed to save revision history to localStorage", e);
        }
        return updatedHistory;
    });
  }, []);

  const handleRestoreRevision = useCallback((revisionCode: string) => {
    const oldCode = code;
    setCode(revisionCode);
    addRevision("Restored from history", revisionCode);
  }, [code, addRevision]);

  const handleToggleBreakpoint = useCallback((lineNumber: number) => {
    setBreakpoints(prev => {
        const newBreakpoints = new Set(prev);
        if (newBreakpoints.has(lineNumber)) {
            newBreakpoints.delete(lineNumber);
        } else {
            newBreakpoints.add(lineNumber);
        }
        return newBreakpoints;
    });
  }, []);


  useEffect(() => {
    executionWorkerRef.current = new Worker('/execution.js');
    executionWorkerRef.current.onmessage = (event: MessageEvent<ExecutionMessage>) => {
        const { type, data } = event.data;
        if (type === 'done') {
            setIsExecuting(false);
        } else if (type === 'stdout') {
            setExecutionOutput(prev => [...prev, { type: 'stdout', message: data }]);
        } else if (type === 'stderr' || type === 'error') {
            setExecutionOutput(prev => [...prev, { type: 'stderr', message: data }]);
        }
    };

    return () => {
        executionWorkerRef.current?.terminate();
    }
  }, []);

  const isConfigured = (s: Settings) => {
    return s.provider === 'gemini' ? !!process.env.API_KEY : !!s.localAIUrl;
  }

  const handleExecuteCode = useCallback(() => {
    if (executionWorkerRef.current) {
        addRevision("Code executed", code);
        setIsExecuting(true);
        setExecutionOutput([]);
        // TODO: Pass breakpoints to worker for debugging
        executionWorkerRef.current.postMessage({ code, breakpoints: Array.from(breakpoints) });
    }
  }, [code, addRevision, breakpoints]);

  const handleAnalyze = useCallback(async () => {
    if (!isConfigured(settings)) {
        setError("AI provider is not configured. Please add an API key or a local URL in the settings.");
        return;
    }
    setIsLoading(true);
    setError(null);
    setSuggestions(null);
    setOriginalSelection('');
    setLastAction(null);
    try {
      const result = await analyzeCode(code, settings);
      setSuggestions(result);
    } catch (err: any) {
      setError(err.message || 'Failed to get suggestions. Please check the console for more details.');
      console.error(err);
    } finally {
      setIsLoading(false);
    }
  }, [code, settings]);

  const handleContextualAnalysis = useCallback(async (selectedCode: string, action: AIAction) => {
    if (!selectedCode) return;
    if (!isConfigured(settings)) {
        setError("AI provider is not configured. Please add an API key or a local URL in the settings.");
        return;
    }

    setIsLoading(true);
```

```
    setError(null);
    setSuggestions(null);
    setOriginalSelection(selectedCode);
    setLastAction(action);
    try {
        const result = await analyzeSelection(selectedCode, action, settings);
        setSuggestions(result);
    } catch (err: any) {
        setError(err.message || 'Failed to get suggestions for the selected code. Please check the console.');
        console.error(err);
    } finally {
        setIsLoading(false);
    }
  }, [settings]);

  const handleAnalyzeDependencies = useCallback(async () => {
    if (!isConfigured(settings)) {
        setError("AI provider is not configured. Please add an API key or a local URL in the settings.");
        return;
    }
    setIsAnalyzingDependencies(true);
    setError(null);
    setRequirementsFile(null);
    try {
        const result = await analyzeDependencies(code, settings);
        setDependencies(result);
    } catch (err: any) {
        setError(err.message || 'Failed to analyze dependencies.');
        console.error(err);
    } finally {
        setIsAnalyzingDependencies(false);
    }
  }, [code, settings]);

  const handleGenerateRequirements = useCallback(async () => {
    if (!dependencies) return;
    setIsAnalyzingDependencies(true); // Reuse loading state
    setError(null);
    try {
        const result = await generateRequirementsFile(dependencies, settings);
        setRequirementsFile(result);
    } catch (err: any) {
        setError(err.message || 'Failed to generate requirements.txt.');
        console.error(err);
    } finally {
        setIsAnalyzingDependencies(false);
    }
  }, [dependencies, settings]);

  const handleExplainError = useCallback(async (traceback: string) => {
    if (!isConfigured(settings)) {
        setError("AI provider is not configured. Please add an API key or a local URL in the settings.");
        return;
    }
    setIsLoading(true);
    setError(null);
    setSuggestions(null);
    setOriginalSelection(code); // The context for the error is the whole script
    setLastAction('explain_error');
    try {
        const result = await explainError(code, traceback, settings);
        setSuggestions(result);
    } catch (err: any) {
        setError(err.message || 'Failed to get an explanation for the error.');
        console.error(err);
    } finally {
        setIsLoading(false);
    }
  }, [code, settings]);
```

```
const handleAcceptChange = (newCode: string) => {
  let finalCode: string;
  let summary: string;

  if (lastAction === 'explain_error') {
      finalCode = newCode;
      summary = "Applied AI fix for error";
  } else {
      finalCode = code.replace(originalSelection, newCode);
      summary = `Accepted AI suggestion for '${lastAction}'`;
  }
  setCode(finalCode);
  addRevision(summary, finalCode);
  setSuggestions(null);
  setOriginalSelection('');
  setLastAction(null);
};

const handleRejectChange = () => {
  setSuggestions(null);
  setOriginalSelection('');
  setLastAction(null);
}

const [selectedFileContent, setSelectedFileContent] = useState<string>("");

const handleFileSelect = (fileContent: string) => {
    setSelectedFileContent(fileContent);
};

return (
  <div className="h-screen flex flex-col bg-background text-text-primary overflow-hidden">
    <Header />
    <FileGraph />
    <div className="app-container flex-grow flex flex-col p-4 lg:p-6 gap-4 overflow-hidden">
      <Sidebar onFileSelect={handleFileSelect} />
      <main className="main-content flex-grow grid grid-cols-1 lg:grid-cols-2 gap-4 overflow-auto">
        {selectedFileContent ? (
          <MarkdownViewer fileContent={selectedFileContent} />
        ) : (
          <p>Select a file to view its content.</p>
        )}
        <CodeDisplay
            code={code}
            onCodeChange={setCode}
            onContextualAnalyze={handleContextualAnalysis}
            onAnalyzeDependencies={handleAnalyzeDependencies}
            onExecuteCode={handleExecuteCode}
            isExecuting={isExecuting}
            breakpoints={breakpoints}
            onToggleBreakpoint={handleToggleBreakpoint}
        />
        <SuggestionsDisplay
          suggestions={suggestions}
          originalCode={originalSelection}
          isLoading={isLoading}
          error={error}
          onAcceptChange={handleAcceptChange}
          onRejectChange={handleRejectChange}
          lastAction={lastAction}
          dependencies={dependencies}
          isAnalyzingDependencies={isAnalyzingDependencies}
          onGenerateRequirements={handleGenerateRequirements}
          requirementsFile={requirementsFile}
          revisionHistory={revisionHistory}
          currentCode={code}
          onRestoreRevision={handleRestoreRevision}
        />
      </main>
      <ExecutionConsole
```

```
            output={executionOutput}
            isExecuting={isExecuting}
            onClear={() => setExecutionOutput([])}
            onExplainError={handleExplainError}
        />
      </div>
      <footer className="bg-background/80 backdrop-blur-sm p-4 border-t border-surface">
        <div className="max-w-7xl mx-auto flex justify-center">
            <AnalyzeButton onClick={handleAnalyze} isLoading={isLoading} />
        </div>
      </footer>
    </div>
  );
};

const App: React.FC = () => {
  return (
    <SettingsProvider>
      <ThemeProvider>
        <AppContent />
      </ThemeProvider>
    </SettingsProvider>
  )
}

export default App;
```

README.md

```
# ■ AI Code Refactor IDE

Welcome to the AI Code Refactor IDE, a next-generation, browser-based development environment designed to supercharge y

This is not just a text editor; it's an intelligent partner that helps you write better, cleaner, and more efficient Py

![AI Code Refactor Screenshot](https://storage.googleapis.com/aistudio-hosting/generative_ai_studio/images/screenshot.p

## ■ Core Features

*   **Comprehensive AI Analysis:** Get a full-script review from an AI expert that covers code quality, best practices
*   **Contextual "Magic Wand" Actions:** Select any piece of code and get instant, targeted help:
    *   **Explain Code:** Understand complex snippets in plain English.
    *   **Refactor for Readability:** Improve code clarity and maintainability.
    *   **Add Docs & Types:** Automatically generate PEP 257 docstrings and PEP 484 type hints.
    *   **Find Potential Bugs:** Let the AI act as a QA engineer to find edge cases and errors.
    *   **Generate Unit Tests:** Create a full `unittest` test suite for your functions and classes in a single click.
*   **Side-by-Side Diff Viewer:** Clearly see the AI's suggested changes and accept or reject them with one click.
*   **Real-time Linting:** Get instant feedback on syntax errors and style issues as you type, powered by the high-perf
*   **AI-Powered Dependency Analysis:** Automatically scan your `import` statements, identify third-party libraries, an
*   **In-Browser Code Execution:** Run your Python scripts directly in the IDE and see the output in an integrated cons
*   **Customizable UI:** Choose from multiple themes to personalize your coding environment.
*   **Flexible AI Provider:** Configure the IDE to use Google's Gemini API or your own local AI model endpoint.

## ■ How to Use

1.  **Configuration (First-Time Use):**
    *   Click the **Settings** (■■) icon in the top-right corner.
    *   Select your AI Provider (Google Gemini is the default).
    *   If using Gemini, your API key should be pre-configured.
    *   Your key is saved securely in your browser's local storage.
2.  **Full Code Analysis:**
    *   Paste your Python code into the editor on the left.
    *   Click the big **"Analyze & Refactor"** button at the bottom.
    *   The AI's detailed feedback will appear in the "AI Suggestions" panel on the right.
3.  **Using the Magic Wand:**
    *   Highlight a function, class, or any snippet of code in the editor.
    *   A ■ wand icon will appear. Click it to open the AI Action Menu.
    *   Select an action like "Refactor" or "Generate Tests".
    *   The results will be displayed on the right, often with a diff view to compare the changes.
4.  **Running Your Code:**
```

```
    *   Click the **"Run Script"** (■■) button in the IDE header.
    *   The output console at the bottom will expand and show any printed messages or errors.
5.  **Managing Dependencies:**
    *   Click the **"Scan Dependencies"** button in the IDE header.
    *   Switch to the "Dependencies" tab on the right-hand panel to see a list of detected libraries.
    *   Click **"Generate requirements.txt"** to create the file content.

## ■■ Technology Stack

*   **Frontend:** React, TypeScript, Tailwind CSS
*   **AI Integration:** `@google/genai` for the Gemini API
*   **In-Browser Python:** [Pyodide](https://pyodide.org/) (powers both the Ruff linter and the code execution engine)
*   **Syntax Highlighting:** Prism.js
*   **Code Editor:** `react-simple-code-editor`

This project is a demonstration of how modern web technologies and powerful AI APIs can be combined to create sophistic
```

ROADMAP.md

```
# ■■ Project Roadmap

This document outlines the development roadmap for the AI Code Refactor IDE. Our vision is to create a fully-featured,

## ■ Completed Milestones

We have successfully built the foundational pillars of the IDE.

*   **V1 - Core AI Assistant**
    *   [x] **Full-Script AI Analysis:** Comprehensive code review via the Gemini API.
    *   [x] **Contextual "Magic Wand" Actions:**
        *   [x] Explain Code
        *   [x] Refactor Code
        *   [x] Add Docs & Type Hints
        *   [x] Find Potential Bugs
    *   [x] **Interactive Diff Viewer:** For reviewing and accepting/rejecting AI suggestions.
    *   [x] **Configurable AI Provider:** Support for both Gemini cloud API and local AI models.
    *   [x] **UI Theming:** Multiple color schemes for personalization.

*   **V2 - The "Integrated" IDE**
    *   [x] **Real-time Linting:** Integrated Ruff linter via a Pyodide web worker for non-blocking diagnostics.
    *   [x] **AI Dependency Scanner:** Intelligent identification of third-party packages.
    *   [x] **`requirements.txt` Generation:** One-click creation of dependency files.
    *   [x] **AI Unit Test Generation:** Automated creation of `unittest` test suites.
    *   [x] **In-Editor Code Execution:** A full Python runtime (Pyodide) with an integrated output console.
    *   [x] **AI-Assisted Debugging:** Automated root cause analysis for runtime errors.


## ■ What's Next? The Path Forward

Our focus now shifts to making the IDE more powerful for real-world projects, enhancing the debugging experience, and e

### ■ V3 - The Debugging & Version Control Powerhouse

*   **Version Control Foundations:**
    *   [x] **Local Revision History:** Automatically save and restore code snapshots.
*   **Advanced Debugging:**
    *   **Intelligent Breakpoints:** Allow setting breakpoints in the editor. (In Progress)
    *   **AI State Inspection:** When paused, use AI to explain the current state of variables (`locals()`, `globals()`
*   **Full Git Integration:**
    *   **Connect to GitHub:** Authenticate with GitHub to clone, view, and edit files from your repositories.
    *   **In-App Staging & Diffing:** View file changes (`git diff`) directly in the IDE.
    *   **AI-Generated Commit Messages:** Automatically generate conventional commit messages based on staged changes.
    *   **Push to Repo:** Commit and push changes back to GitHub without leaving the application.

### ■ V4 - The Collaborative & Extensible IDE

*   **Multi-File Support:**
    *   **File System Explorer:** A sidebar to navigate the file and directory structure of a project.
    *   **Tabs for Open Files:** Manage multiple open files just like a desktop IDE.
    *   **Project-Wide AI Actions:** Perform analysis and refactoring across the entire codebase.
*   **Collaborative Coding:**
    *   **Real-time Multiplayer Editing:** Similar to Google Docs, allow multiple users to code in the same session.
    *   **AI Pair Programmer Mode:** Have the AI act as an active participant in the session, offering suggestions and
*   **Extension API:**
    *   **User-Defined AI Actions:** Create a simple framework for users to add their own "Magic Wand" actions with cus
    *   **Plugin Marketplace:** A long-term vision to allow users to share the extensions they've built.

This roadmap is ambitious, but it charts a clear path toward redefining what a web-based IDE can be.
```

UPGRADEME.md

```
# ■■ UPGRADEME.md - Developer & Contributor Guide

Welcome, developer! This guide is for you if you're interested in extending, modifying, or simply understanding the inn

## ■■ Project Architecture Overview

The application is a client-side single-page application (SPA) built with React and TypeScript. There is no server-side

*   **`App.tsx`**: The main component that manages the application's state, including the code, AI suggestions, depende
*   **`components/`**: Contains all the React components that make up the UI. Each component is designed to be as self-
*   **`services/geminiService.ts`**: This is the heart of the AI integration. It contains the logic for constructing pr
*   **`contexts/`**: React Context providers for managing global state like Settings (`SettingsContext.tsx`) and UI The
*   **`public/`**: Contains static assets, including the web workers (`linter.js`, `execution.js`) that run Python and

##  Extending the IDE

Here are guides for common extension tasks.

### How to Add a New "Magic Wand" AI Action

Adding a new action is a three-step process:

1.  **Define the Action in `geminiService.ts`:**
    *   Open `services/geminiService.ts`.
    *   Add your new action's ID to the `AIAction` type alias. For example: `export type AIAction = 'explain' | 'refact
    *   In the `getPromptForAction` function, add a new `case` for your action. Here, you will define the prompt you wa

2.  **Add the UI Button in `AIActionMenu.tsx`:**
    *   Open `components/AIActionMenu.tsx`.
    *   Add a new icon for your action to `components/Icon.tsx` if needed.
    *   In the `actions` array, add a new object for your action, including its `id` (which must match the one in `gemi

3.  **Handle the Response (if necessary):**
    *   If your new action returns a unique response structure (i.e., not a standard diff or test case), you may need t

### How to Add a New Theme

1.  **Define the Theme in `constants/themes.ts`:**
    *   Open `constants/themes.ts`.
    *   Add a new `Theme` object to the `themes` array.
    *   Give it a unique `name` and define the six CSS variable colors. You can use any valid CSS color format (hex, rg

2.  **That's it!** The `SettingsModal.tsx` component will automatically pick up your new theme and list it in the "Appe

### How to Update In-Browser Python Packages

The Python environment runs via Pyodide. The packages available (like `ruff`) are installed within the web workers.

*   **To update Ruff:** In `public/linter.js`, find the line `await micropip.install("ruff");` and change it to `await
*   **To add a new package:** If you need another package for execution or linting, you can add another `micropip.insta
```

constants.ts

```
export const PYTHON_CODE = `#!/usr/bin/env python3
"""
■ LYRA - Ultimate Codex PR Automation Beast
Auto-resolves conflicts, merges PRs, cleans up branches. ONE FILE TO RULE THEM ALL!
"""
```

```
import os
import sys
import time
import logging
import argparse
import yaml
import json
import requests
from datetime import datetime, timezone
from typing import List, Optional, Tuple, Dict, Any
import re
import tempfile
import shutil
from pathlib import Path

# External dependencies with fallbacks
try:
    from github import Github, GithubException, Auth
    from github.PullRequest import PullRequest as PRType
    from github.Repository import Repository
    from github.GitCommit import GitCommit
    from github.Issue import Issue
    from git import Repo
    HAS_GIT = True
except ImportError as e:
    print(f"Missing dependency: {e}")
    print("Install with: pip install PyGithub gitpython PyYAML python-dotenv tenacity requests")
    sys.exit(1)

try:
    from dotenv import load_dotenv
    load_dotenv()
except ImportError:
    pass

try:
    from tenacity import retry, stop_after_attempt, wait_exponential, retry_if_exception_type
except ImportError:
    retry = lambda x: x
    logger = logging.getLogger(__name__)
    logger.warning("tenacity not installed; rate-limit retries disabled.")

# ================================
# LOGGING SETUP
# ================================
def setup_logging(verbose: bool = False, log_path: str = "lyra.log") -> None:
    level = logging.DEBUG if verbose else logging.INFO
    handlers = [logging.StreamHandler(), logging.FileHandler(log_path)]
    logging.basicConfig(
        level=level,
        format="[%(asctime)s] %(levelname)s: %(message)s",
        datefmt="%Y-%m-%d %H:%M:%S",
        handlers=handlers
    )

logger = logging.getLogger(__name__)

# ================================
# CONFIG & ENV
# ================================
def load_config(config_path: str = "lyra.yaml") -> Dict:
    defaults = {
        "merge_method": os.getenv("MERGE_METHOD", "squash"),
        "max_files": int(os.getenv("MAX_FILES", "50")),
        "max_attempts": int(os.getenv("MAX_ATTEMPTS", "3")),
        "conflict_strategy": os.getenv("CONFLICT_STRATEGY", "ours"),
        "kobold_url": os.getenv("KOBOLD_URL", "http://localhost:5001/api/v1/generate"),
        "slack_webhook": os.getenv("SLACK_WEBHOOK"),
        "label": os.getenv("CODEX_LABEL", "codex"),
        "auto_approve": os.getenv("AUTO_APPROVE", "true").lower() == "true"
    }

    try:
        if os.path.exists(config_path):
            with open(config_path) as f:
                config = yaml.safe_load(f) or {}
            defaults.update(config.get("global", {}))
    except Exception as e:
        logger.warning(f"Failed to load {config_path}: {e}")

    return defaults

def getenv(name: str, default: Optional[str] = None, required: bool = False) -> str:
    value = os.getenv(name, default)
    if required and not value:
        logger.error(f"Missing required env var: {name}")
        sys.exit(1)
    return value

def human_int(n: int) -> str:
    return f"{n:,}"

def load_cache(cache_path: str = "lyra_cache.json") -> Dict:
    try:
        if os.path.exists(cache_path):
            with open(cache_path) as f:
                return json.load(f)
    except:
        pass
    return {}

def save_cache(cache: Dict, cache_path: str = "lyra_cache.json") -> None:
    try:
        with open(cache_path, "w") as f:
            json.dump(cache, f, indent=2)
    except Exception as e:
        logger.warning(f"Failed to save cache: {e}")

# ================================
# AI INTEGRATION
# ================================
def query_ai(prompt: str, kobold_url: str) -> Optional[str]:
    try:
        payload = {"prompt": prompt, "max_length": 500, "temperature": 0.7, "top_p": 0.9}
        resp = requests.post(kobold_url, json=payload, timeout=15)
        resp.raise_for_status()
        return resp.json().get("results", [{}])[0].get("text", "").strip()
    except:
        return None

def ai_analyze_conflict(repo: Repository, pr: PRType, diff: str) -> str:
    prompt = f"""Analyze this GitHub PR conflict in {repo.full_name}:

PR #{pr.number}: {pr.title}
Status: CONFLICTED

{diff[:2000]}

RECOMMENDATION (ours/theirs/manual):"""

    suggestion = query_ai(prompt, load_config()["kobold_url"])
    return suggestion or "ours"

# ================================
# CONFLICT RESOLUTION BEAST ■
# ================================
@retry(stop=stop_after_attempt(3), wait=wait_exponential(min=2, max=10))
def resolve_pr_conflicts(repo: Repository, pr: PRType, config: Dict) -> Tuple[bool, str, Dict]:
    """The heart of Lyra - auto-resolves conflicts like magic ■"""
    if pr.mergeable_state != "conflicted":
        return True, "No conflicts", {"resolved": 0}

    stats = {"resolved": 0, "files": 0, "strategy": config["conflict_strategy"]}

    logger.info(f"■ RESOLVING CONFLICTS → PR #{pr.number}")

    with tempfile.TemporaryDirectory() as temp_dir:
        try:
            # Clone repo
            token = getenv("GITHUB_TOKEN", required=True)
            url = f"https://{token}@github.com/{repo.full_name}.git"
            local_repo = Repo.clone_from(url, temp_dir)

            base = pr.base.ref
            head = pr.head.ref

            # Setup for merge
            local_repo.git.checkout(base)
            local_repo.git.fetch(f"origin", f"pull/{pr.number}/head:{head}")

            # AI-powered strategy decision
            files = list(pr.get_files())
            diff = "\\n".join([f.file_to_patch() for f in files[:3]])
            strategy = ai_analyze_conflict(repo, pr, diff)
            logger.info(f"■ AI suggests: {strategy}")

            # Execute merge with strategy
            merge_args = [head]
            if strategy == "theirs":
                merge_args.extend(["--strategy-option=theirs"])
            else:  # ours or default
                merge_args.extend(["--strategy-option=ours"])

            local_repo.git.merge(*merge_args)

            # Create fixed branch
            ts = int(time.time())
            fixed_branch = f"lyra-fixed/pr-{pr.number}-{ts}"
            local_repo.git.checkout("-b", fixed_branch)
            local_repo.git.push("origin", fixed_branch, force=True)

            # Update PR
            pr.edit(head=fixed_branch)
            pr.create_issue_comment(
                f"■ **LYRA CONFLICTS RESOLVED!**\\n\\n"
                f"■ Strategy: \`{strategy}\`\\n"
                f"■ Fixed branch: \`{fixed_branch}\`\\n"
                f"■ **READY TO MERGE**"
            )

            stats["resolved"] = 1
            stats["files"] = len(files)

            logger.info(f"■ FIXED PR #{pr.number} → {fixed_branch}")
            return True, f"Fixed: {fixed_branch}", stats

        except Exception as e:
            logger.error(f"■ Conflict resolution failed PR #{pr.number}: {e}")
            return False, str(e), stats

# ================================
# MERGE + CLEANUP
# ================================
def try_merge_pr(repo: Repository, pr: PRType, config: Dict, pr_cache: Dict) -> Tuple[bool, str, Dict]:
    """Attempt merge with full conflict resolution pipeline"""

    stats = {"merged": 0, "conflicts": 0, "reviews": 0}
    cache_key = f"{repo.full_name}/pr/{pr.number}"
    # Conflict resolution first
    if not pr.mergeable and pr.mergeable_state == "conflicted":
        success, msg, conflict_stats = resolve_pr_conflicts(repo, pr, config)
        stats.update(conflict_stats)

        if not success:
            return False, f"Conflict resolution failed: {msg}", stats

        # Refresh PR after resolution
        time.sleep(2)
        pr = repo.get_pull(pr.number)

    if not pr.mergeable:
        return False, f"Still not mergeable: {pr.mergeable_state}", stats

    # Auto-approve if enabled
    if config["auto_approve"]:
        reviews = list(pr.get_reviews())
        if len([r for r in reviews if r.state == "APPROVED"]) == 0:
            try:
                pr.create_review(event="APPROVE", body="■ Lyra Auto-Approval")
                stats["reviews"] = 1
            except:
                pass

    # Merge!
    try:
        method = config["merge_method"]
        if method == "squash":
            pr.merge(merge_method="squash")
        elif method == "rebase":
            pr.merge(merge_method="rebase")
        else:
            pr.merge()

        stats["merged"] = 1

        # CLEANUP BRANCHES ■
        cleanup_branches(repo, pr)

        msg = f"■ MERGED #{pr.number}: {pr.title}"
        logger.info(msg)

        # Notify
        pr.create_issue_comment(f"■ **MERGED BY LYRA!** ■")
        send_slack(f"■ {repo.full_name} PR #{pr.number} MERGED!")

        return True, msg, stats

    except Exception as e:
        return False, f"Merge failed: {str(e)}", stats

def cleanup_branches(repo: Repository, pr: PRType) -> None:
    """Delete source branch + all lyra-fixed branches"""
    try:
        # Original branch
        if pr.head.repo.full_name == repo.full_name:
            repo.delete_git_ref(f"heads/{pr.head.ref}")

        # All lyra-fixed branches for this PR
        branches = [b.name for b in repo.get_branches()]
        pattern = re.compile(rf"lyra-fixed/pr-{pr.number}-\\d+")

        for branch in branches:
            if pattern.match(branch):
                repo.delete_git_ref(f"heads/{branch}")
                logger.info(f"■ Deleted: {branch}")

    except Exception as e:
        logger.warning(f"Cleanup failed: {e}")

# ================================
# NOTIFICATIONS
# ================================
def send_slack(message: str) -> None:
    webhook = load_config()["slack_webhook"]
    if not webhook:
        return
    try:
        requests.post(webhook, json={"text": f"■ *Lyra*\\n{message}"})
    except:
        pass

# ================================
# MAIN PROCESSING ENGINE
# ================================
@retry(stop=stop_after_attempt(3), wait=wait_exponential(min=4, max=10))
def process_repo(repo_full: str, label: str, config: Dict, pr_cache: Dict) -> str:
    """Process single repo - THE CORE LOGIC"""

    g = Github(auth=Auth.Token(getenv("GITHUB_TOKEN")))
    repo = g.get_repo(repo_full)

    if repo.archived:
        return f"[{repo_full}] Skipped (archived)"

    # Fetch PRs
    prs = list(repo.get_pulls(state="open", sort="updated", direction="desc"))
    codex_prs = []

    for pr in prs:
        title = pr.title.lower()
        labels = {l.name.lower() for l in pr.get_labels()}

        if label.lower() in labels or label.lower() in title:
            codex_prs.append(pr)

    if not codex_prs:
        return f"[{repo_full}] No codex PRs"

    stats = {"merged": 0, "conflicts": 0, "updated": 0, "blocked": 0}
    notes = []

    for pr in codex_prs:
        ok, msg, pr_stats = try_merge_pr(repo, pr, config, pr_cache)
        stats.update(pr_stats)

        if ok:
            stats["merged"] += 1
        elif "conflict" in msg.lower():
            stats["conflicts"] += 1
        elif "behind" in msg.lower():
            stats["updated"] += 1
        else:
            stats["blocked"] += 1

        notes.append(msg)

    summary = (
        f"[{repo_full}] PRs: {len(codex_prs)} | "
        f"■{stats['merged']} ■{stats['conflicts']} ■{stats['updated']} "
        f"■{stats['blocked']}"
    )

    logger.info(f"{summary} | {' | '.join(notes)}")
    return summary

# ================================
# ORCHESTRATOR
# ================================
def process_all_repos(label: str, dry_run: bool, config: Dict) -> str:
    """Process ALL repos"""
    if dry_run:
        logger.info("■ DRY RUN MODE")
        return "DRY RUN COMPLETE"

    g = Github(auth=Auth.Token(getenv("GITHUB_TOKEN")))
    org = os.getenv("ORG")

    if org:
        target = g.get_organization(org)
    else:
        target = g.get_user()

    repos = [r.full_name for r in target.get_repos() if not r.archived]
    pr_cache = load_cache()
    summaries = []

    for repo_full in repos:
        try:
            summary = process_repo(repo_full, label, config, pr_cache)
            summaries.append(summary)
        except Exception as e:
            summaries.append(f"[{repo_full}] ERROR: {e}")

    save_cache(pr_cache)
    final = f"■ LYRA COMPLETE: {len([s for s in summaries if '■' in s])} merged!"
    logger.info(final)

    return "\\n".join(summaries)

# ================================
# CLI + MAIN
# ================================
def main():
    parser = argparse.ArgumentParser(description="■ Lyra - Codex PR Automation Beast")
    parser.add_argument("--repo", help="Single repo (org/repo)")
    parser.add_argument("--label", default="codex", help="PR label/keyword")
    parser.add_argument("--interval", type=int, default=0, help="Run every N seconds")
    parser.add_argument("--dry-run", action="store_true", help="Simulate only")
    parser.add_argument("--verbose", "-v", action="store_true")
    parser.add_argument("--config", default="lyra.yaml")

    args = parser.parse_args()

    setup_logging(args.verbose)
    config = load_config(args.config)

    def run_once():
        if args.repo:
            summary = process_repo(args.repo, args.label, config, {})
        else:
            summary = process_all_repos(args.label, args.dry_run, config)
        print(summary)
        send_slack(summary)

    if args.interval == 0:
        run_once()
    else:
        logger.info(f"■ Lyra watching every {args.interval}s | Label: {args.label}")
        while True:
            try:
                run_once()
                time.sleep(args.interval)
            except KeyboardInterrupt:
                logger.info("■ Lyra stopped")
                break

if __name__ == "__main__":
    main()
`;
```

file_links.json

```
{
  "files": [
    {
      "name": ".env.local",
      "type": "file"
    },
    {
      "name": ".gitignore",
      "type": "file"
    },
    {
      "name": "App.tsx",
      "type": "file"
    },
    {
      "name": "components",
      "type": "folder"
    },
    {
      "name": "constants",
      "type": "folder"
    },
    {
      "name": "constants.ts",
      "type": "file"
    },
    {
      "name": "contexts",
      "type": "folder"
    },
    {
      "name": "file_links.json",
      "type": "file"
    },
    {
      "name": "generate_file_links.cjs",
      "type": "file"
    },
    {
      "name": "index.html",
      "type": "file"
    },
    {
      "name": "index.tsx",
      "type": "file"
    },
    {
      "name": "metadata.json",
      "type": "file"
    },
    {
      "name": "node_modules",
      "type": "folder"
    },
    {
      "name": "package-lock.json",
      "type": "file"
    },
    {
      "name": "package.json",
      "type": "file"
    },
    {
      "name": "public",
      "type": "folder"
    },
    {
      "name": "README.md",
      "type": "file"
    },
    {
      "name": "ROADMAP.md",
      "type": "file"
    },
    {
      "name": "services",
      "type": "folder"
    },
    {
      "name": "tsconfig.json",
      "type": "file"
    },
    {
      "name": "UPGRADEME.md",
      "type": "file"
    },
    {
      "name": "vite.config.ts",
      "type": "file"
    }
  ]
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>AI Code Refactor</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <script>
      tailwind.config = {
        theme: {
          extend: {
            colors: {
              'primary': 'var(--color-primary)',
              'secondary': 'var(--color-secondary)',
              'background': 'var(--color-background)',
              'surface': 'var(--color-surface)',
              'text-primary': 'var(--color-text-primary)',
              'text-secondary': 'var(--color-text-secondary)',
            },
            fontFamily: {
                sans: ['Inter', 'sans-serif'],
                mono: ['Fira Code', 'monospace'],
            }
          }
        }
      }
    </script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Fira+Code:wght@400;500;700&family=Inter:wght@400;500;600;700&d
    <link href="https://cdnjs.cloudflare.com/ajax/libs/prism/1.29.0/themes/prism-tomorrow.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/pyodide/v0.25.1/full/pyodide.js"></script>
  <script type="importmap">
{
  "imports": {
    "react-dom/": "https://aistudiocdn.com/react-dom@^19.2.0/",
    "react/": "https://aistudiocdn.com/react@^19.2.0/",
    "react": "https://aistudiocdn.com/react@^19.2.0",
    "@google/genai": "https://aistudiocdn.com/@google/genai@^1.27.0",
    "react-simple-code-editor": "https://esm.sh/react-simple-code-editor@0.13.1",
    "prismjs": "https://esm.sh/prismjs@1.29.0",
    "prismjs/": "https://esm.sh/prismjs@1.29.0/"
  }
}
</script>
<link rel="stylesheet" href="/index.css">
</head>
  <body class="bg-background text-text-primary font-sans">
    <div id="root"></div>
    <script type="module" src="/index.tsx"></script>
  </body>
</html>
```

index.tsx

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

const rootElement = document.getElementById('root');
if (!rootElement) {
  throw new Error("Could not find root element to mount to");
}

const root = ReactDOM.createRoot(rootElement);
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

metadata.json

```
{
  "name": "EchoForge",
  "description": "An application that uses the Gemini API to analyze Python code, providing detailed suggestions for im
  "requestFramePermissions": []
}
```

package-lock.json

```
{
  "name": "echoforge",
  "version": "0.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "echoforge",
      "version": "0.0.0",
      "dependencies": {
        "@google/genai": "^1.27.0",
        "prismjs": "^1.30.0",
        "react": "^19.2.0",
        "react-dom": "^19.2.0",
        "react-markdown": "^8.0.0",
        "react-simple-code-editor": "0.13.1"
      },
      "devDependencies": {
        "@types/node": "^22.14.0",
        "@types/react": "^19.2.6",
        "@vitejs/plugin-react": "^5.0.0",
        "typescript": "~5.8.2",
        "vite": "^6.2.0"
      }
    },
    "node_modules/@babel/code-frame": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/code-frame/-/code-frame-7.27.1.tgz",
      "integrity": "sha512-cjQ7ZlQ0Mv3b47hABuTevyTuYN4i+loJKGeV9flcCgIK37cCXRh+L1bd3iBHlynerhQ7BhCkn2BPbQUL+rGqFg==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/helper-validator-identifier": "^7.27.1",
        "js-tokens": "^4.0.0",
        "picocolors": "^1.1.1"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/compat-data": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/compat-data/-/compat-data-7.28.5.tgz",
      "integrity": "sha512-6uFXyCayocRbqhZOB+6XcuZbkMNimwfVGFji8CTZnCzOHVGvDqzvitu1re2AU5LROliz7eQPhB8CpAMvnx9EjA==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/core": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/core/-/core-7.28.5.tgz",
      "integrity": "sha512-e7jT4DxYvIDLk1ZHmU/m/mB19rex9sv0c2ftBtjSBv+kVM/902eh0fINUzD7UwLLNR+jU585GxUJ8/EBfAM5fw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/code-frame": "^7.27.1",
        "@babel/generator": "^7.28.5",
        "@babel/helper-compilation-targets": "^7.27.2",
        "@babel/helper-module-transforms": "^7.28.3",
        "@babel/helpers": "^7.28.4",
        "@babel/parser": "^7.28.5",
        "@babel/template": "^7.27.2",
        "@babel/traverse": "^7.28.5",
        "@babel/types": "^7.28.5",
        "@jridgewell/remapping": "^2.3.5",
        "convert-source-map": "^2.0.0",
        "debug": "^4.1.0",
        "gensync": "^1.0.0-beta.2",
        "json5": "^2.2.3",
        "semver": "^6.3.1"
      },
      "engines": {
        "node": ">=6.9.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/babel"
      }
    },
    "node_modules/@babel/generator": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/generator/-/generator-7.28.5.tgz",
      "integrity": "sha512-3EwLFhZ38J4VyIP6WNtt2kUdW9dokXA9Cr4IVIFHuCpZ3H8/YFOl5JjZHisrn1fATPBmKKqXzDFvh9fUwHz6CQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/parser": "^7.28.5",
        "@babel/types": "^7.28.5",
        "@jridgewell/gen-mapping": "^0.3.12",
        "@jridgewell/trace-mapping": "^0.3.28",
        "jsesc": "^3.0.2"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-compilation-targets": {
      "version": "7.27.2",
      "resolved": "https://registry.npmjs.org/@babel/helper-compilation-targets/-/helper-compilation-targets-7.27.2.tgz
      "integrity": "sha512-2+1thGUUWWjLTYTHZWK1n8Yga0ijBz1XAhUXcKy81rd5g6yh7hGqMp45v7cadSbEHc9G3OTv45SyneRN3ps4DQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/compat-data": "^7.27.2",
        "@babel/helper-validator-option": "^7.27.1",
        "browserslist": "^4.24.0",
        "lru-cache": "^5.1.1",
        "semver": "^6.3.1"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-globals": {
      "version": "7.28.0",
      "resolved": "https://registry.npmjs.org/@babel/helper-globals/-/helper-globals-7.28.0.tgz",
      "integrity": "sha512-+W6cISkXFa1jXsDEdYA8HeevQT/FULhxzR99pxphltZcVaugps53THCeiWA8SguxxpSp3gKPiuYfSWopkLQ4hw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-module-imports": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/helper-module-imports/-/helper-module-imports-7.27.1.tgz",
      "integrity": "sha512-0gSFWUPNXNopqtIPQvlD5WgXYI5GY2kP2cCvoT8kczjbfcfuIljTbcWrulD1CIPIX2gt1wghbDy08yE1p+/r3w==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/traverse": "^7.27.1",
        "@babel/types": "^7.27.1"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-module-transforms": {
      "version": "7.28.3",
      "resolved": "https://registry.npmjs.org/@babel/helper-module-transforms/-/helper-module-transforms-7.28.3.tgz",
      "integrity": "sha512-gytXUbs8k2sXS9PnQptz5o0QnpLL51SwASIORY6XaBKF88nsOT0Zw9szLqlSGQDP/4TljBAD5y98p2U1fqkdsw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/helper-module-imports": "^7.27.1",
        "@babel/helper-validator-identifier": "^7.27.1",
        "@babel/traverse": "^7.28.3"
      },
      "engines": {
        "node": ">=6.9.0"
      },
      "peerDependencies": {
        "@babel/core": "^7.0.0"
      }
    },
    "node_modules/@babel/helper-plugin-utils": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/helper-plugin-utils/-/helper-plugin-utils-7.27.1.tgz",
      "integrity": "sha512-1gn1Up5YXka3YYAHGKpbideQ5Yjf1tDa9qYcgysz+cNCXukyLl6DjPXhD3VRwSb8c0J9tA4b2+rHEZtc6R0tlw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-string-parser": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/helper-string-parser/-/helper-string-parser-7.27.1.tgz",
      "integrity": "sha512-qMlSxKbpRlAridDExk92nSobyDdpPijUq2DW6oDnUqd0iOGxmQjyqhMIihI9+zv4LPyZdRje2cavWPbCbWm3eA==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-validator-identifier": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/helper-validator-identifier/-/helper-validator-identifier-7.28.5.t
      "integrity": "sha512-qSs4ifwzKJSV39ucNjsvc6WVHs6b7S03sOh2OcHF9UHfVPqWWALUsNUVzhSBiItjRZoLHx7nIarVjqKVusUZ1Q==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helper-validator-option": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/helper-validator-option/-/helper-validator-option-7.27.1.tgz",
      "integrity": "sha512-YvjJow9FxbhFFKDSuFnVCe2WxXk1zWc22fFePVNEaWJEu8IrZVlda6N0uHwzZrUM1il7NC9Mlp4MaJYbYd9JSg==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/helpers": {
      "version": "7.28.4",
      "resolved": "https://registry.npmjs.org/@babel/helpers/-/helpers-7.28.4.tgz",
      "integrity": "sha512-HFN59MmQXGHVyYadKLVumYsA9dBFun/ldYxipEjzA4196jpLZd8UjEEBLkbEkvfYreDqJhZxYAWFPtrfhNpj4w==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/template": "^7.27.2",
        "@babel/types": "^7.28.4"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/parser": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/parser/-/parser-7.28.5.tgz",
      "integrity": "sha512-KKBU1VGYR7ORr3At5HAtUQ+TV3SzRCXmA/8OdDZiLDBIZxVyzXuztPjfLd3BV1PRAQGCMWWSHYhL0F8d5uHBDQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/types": "^7.28.5"
      },
      "bin": {
        "parser": "bin/babel-parser.js"
      },
      "engines": {
        "node": ">=6.0.0"
      }
    },
    "node_modules/@babel/plugin-transform-react-jsx-self": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/plugin-transform-react-jsx-self/-/plugin-transform-react-jsx-self-
      "integrity": "sha512-6UzkCs+ejGdZ5mFFC/OCUrv028ab2fp1znZmCZjAOBKiBK2jXD1O+BPSfX8X2qjJ75fZBMSnQn3Rq2mrBJK2mw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/helper-plugin-utils": "^7.27.1"
      },
      "engines": {
        "node": ">=6.9.0"
      },
      "peerDependencies": {
        "@babel/core": "^7.0.0-0"
      }
    },
    "node_modules/@babel/plugin-transform-react-jsx-source": {
      "version": "7.27.1",
      "resolved": "https://registry.npmjs.org/@babel/plugin-transform-react-jsx-source/-/plugin-transform-react-jsx-sou
      "integrity": "sha512-zbwoTsBruTeKB9hSq73ha66iFeJHuaFkUbwvqElnygoNbj/jHRsSeokowZFN3CZ64IvEqcmmkVe89OPXc7ldAw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/helper-plugin-utils": "^7.27.1"
      },
      "engines": {
        "node": ">=6.9.0"
      },
      "peerDependencies": {
        "@babel/core": "^7.0.0-0"
      }
    },
    "node_modules/@babel/template": {
      "version": "7.27.2",
      "resolved": "https://registry.npmjs.org/@babel/template/-/template-7.27.2.tgz",
      "integrity": "sha512-LPDZ85aEJyYSd18/DkjNh4/y1ntkE5KwUHWTiqgRxruuZL2F1yuHligVHLvcHY2vMHXttKFpJn6LwfI7cw7ODw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/code-frame": "^7.27.1",
        "@babel/parser": "^7.27.2",
        "@babel/types": "^7.27.1"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/traverse": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/traverse/-/traverse-7.28.5.tgz",
      "integrity": "sha512-TCCj4t55U90khlYkVV/0TfkJkAkUg3jZFA3Neb7unZT8CPok7iiRfaX0F+WnqWqt7OxhOn0uBKXCw4lbL8W0aQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/code-frame": "^7.27.1",
        "@babel/generator": "^7.28.5",
        "@babel/helper-globals": "^7.28.0",
        "@babel/parser": "^7.28.5",
        "@babel/template": "^7.27.2",
        "@babel/types": "^7.28.5",
        "debug": "^4.3.1"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@babel/types": {
      "version": "7.28.5",
      "resolved": "https://registry.npmjs.org/@babel/types/-/types-7.28.5.tgz",
      "integrity": "sha512-qQ5m48eI/MFLQ5PxQj4PFaprjyCTLI37ElWMmNs0K8Lk3dVeOdNpB3ks8jc7yM5CDmVC73eMVk/trk3fgmrUpA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/helper-string-parser": "^7.27.1",
        "@babel/helper-validator-identifier": "^7.28.5"
      },
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/@esbuild/aix-ppc64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/aix-ppc64/-/aix-ppc64-0.25.12.tgz",
      "integrity": "sha512-Hhmwd6CInZ3dwpuGTF8fJG6yoWmsToE+vYgD4nytZVxcu1ulHpUQRAB1UJ8+N1Am3Mz4+xOByoQoSZf4D+CpkA==",
      "cpu": [
        "ppc64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "aix"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/android-arm": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/android-arm/-/android-arm-0.25.12.tgz",
      "integrity": "sha512-VJ+sKvNA/GE7Ccacc9Cha7bpS8nyzVv0jdVgwNDaR4gDMC/2TTRc33Ip8qrNYUcpkOHUT5OZ0bUcNNVZQ9RLlg==",
      "cpu": [
        "arm"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "android"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/android-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/android-arm64/-/android-arm64-0.25.12.tgz",
      "integrity": "sha512-6AAmLG7zwD1Z159jCKPvAxZd4y/VTO0VkprYy+3N2FtJ8+BQWFXU+OxARIwA46c5tdD9SsKGZ/1ocqBS/gAKHg==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "android"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/android-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/android-x64/-/android-x64-0.25.12.tgz",
      "integrity": "sha512-5jbb+2hhDHx5phYR2By8GTWEzn6I9UqR11Kwf22iKbNpYrsmRB18aX/9ivc5cabcUiAT/wM+YIZ6SG9QO6a8kg==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "android"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/darwin-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/darwin-arm64/-/darwin-arm64-0.25.12.tgz",
      "integrity": "sha512-N3zl+lxHCifgIlcMUP5016ESkeQjLj/959RxxNYIthIg+CQHInujFuXeWbWMgnTo4cp5XVHqFPmpyu9J65C1Yg==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "darwin"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/darwin-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/darwin-x64/-/darwin-x64-0.25.12.tgz",
      "integrity": "sha512-HQ9ka4Kx21qHXwtlTUVbKJOAnmG1ipXhdWTmNXiPzPfWKpXqASVcWdnf2bnL73wgjNrFXAa3yYvBSd9pzfEIpA==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "darwin"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/freebsd-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/freebsd-arm64/-/freebsd-arm64-0.25.12.tgz",
      "integrity": "sha512-gA0Bx759+7Jve03K1S0vkOu5Lg/85dou3EseOGUes8flVOGxbhDDh/iZaoek11Y8mtyKPGF3vP8XhnkDEAmzeg==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "freebsd"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/freebsd-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/freebsd-x64/-/freebsd-x64-0.25.12.tgz",
      "integrity": "sha512-TGbO26Yw2xsHzxtbVFGEXBFH0FRAP7gtcPE7P5yP7wGy7cXK2oO7RyOhL5NLiqTlBh47XhmIUXuGciXEqYFfBQ==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "freebsd"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-arm": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-arm/-/linux-arm-0.25.12.tgz",
      "integrity": "sha512-lPDGyC1JPDou8kGcywY0YILzWlhhnRjdof3UlcoqYmS9El818LLfJJc3PXXgZHrHCAKs/Z2SeZtDJr5MrkxtOw==",
      "cpu": [
        "arm"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-arm64/-/linux-arm64-0.25.12.tgz",
      "integrity": "sha512-8bwX7a8FghIgrupcxb4aUmYDLp8pX06rGh5HqDT7bB+8Rdells6mHvrFHHW2JAOPZUbnjUpKTLg6ECyzvas2AQ==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-ia32": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-ia32/-/linux-ia32-0.25.12.tgz",
      "integrity": "sha512-0y9KrdVnbMM2/vG8KfU0byhUN+EFCny9+8g202gYqSSVMonbsCfLjUO+rCci7pM0WBEtz+oK/PIwHkzxkyharA==",
      "cpu": [
        "ia32"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-loong64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-loong64/-/linux-loong64-0.25.12.tgz",
      "integrity": "sha512-h///Lr5a9rib/v1GGqXVGzjL4TMvVTv+s1DPoxQdz7l/AYv6LDSxdIwzxkrPW438oUXiDtwM10o9PmwS/6Z0Ng==",
      "cpu": [
        "loong64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-mips64el": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-mips64el/-/linux-mips64el-0.25.12.tgz",
      "integrity": "sha512-iyRrM1Pzy9GFMDLsXn1iHUm18nhKnNMWscjmp4+hpafcZjrr2WbT//d20xaGljXDBYHqRcl8HnxbX6uaA/eGVw==",
      "cpu": [
        "mips64el"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-ppc64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-ppc64/-/linux-ppc64-0.25.12.tgz",
      "integrity": "sha512-9meM/lRXxMi5PSUqEXRCtVjEZBGwB7P/D4yT8UG/mwIdze2aV4Vo6U5gD3+RsoHXKkHCfSxZKzmDssVlRj1QQA==",
      "cpu": [
        "ppc64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-riscv64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-riscv64/-/linux-riscv64-0.25.12.tgz",
      "integrity": "sha512-Zr7KR4hgKUpWAwb1f3o5ygT04MzqVrGEGXGLnj15YQDJErYu/BGg+wmFlIDOdJp0PmB0lLvxFIOXZgFRrdjR0w==",
      "cpu": [
        "riscv64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-s390x": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-s390x/-/linux-s390x-0.25.12.tgz",
      "integrity": "sha512-MsKncOcgTNvdtiISc/jZs/Zf8d0cl/t3gYWX8J9ubBnVOwlk65UIEEvgBORTiljloIWnBzLs4qhzPkJcitIzIg==",
      "cpu": [
        "s390x"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/linux-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/linux-x64/-/linux-x64-0.25.12.tgz",
      "integrity": "sha512-uqZMTLr/zR/ed4jIGnwSLkaHmPjOjJvnm6TVVitAa08SLS9Z0VM8wIRx7gWbJB5/J54YuIMInDquWyYvQLZkgw==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/netbsd-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/netbsd-arm64/-/netbsd-arm64-0.25.12.tgz",
      "integrity": "sha512-xXwcTq4GhRM7J9A8Gv5boanHhRa/Q9KLVmcyXHCTaM4wKfIpWkdXiMog/KsnxzJ0A1+nD+zoecuzqPmCRyBGjg==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "netbsd"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/netbsd-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/netbsd-x64/-/netbsd-x64-0.25.12.tgz",
      "integrity": "sha512-Ld5pTlzPy3YwGec4OuHh1aCVCRvOXdH8DgRjfDy/oumVovmuSzWfnSJg+VtakB9Cm0gxNO9BzWkj6mtO1FMXkQ==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "netbsd"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/openbsd-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/openbsd-arm64/-/openbsd-arm64-0.25.12.tgz",
      "integrity": "sha512-fF96T6KsBo/pkQI950FARU9apGNTSlZGsv1jZBAlcLL1MLjLNIWPBkj5NlSz8aAzYKg+eNqknrUJ24QBybeR5A==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "openbsd"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/openbsd-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/openbsd-x64/-/openbsd-x64-0.25.12.tgz",
      "integrity": "sha512-MZyXUkZHjQxUvzK7rN8DJ3SRmrVrke8ZyRusHlP+kuwqTcfWLyqMOE3sScPPyeIXN/mDJIfGXvcMqCgYKekoQw==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "openbsd"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/openharmony-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/openharmony-arm64/-/openharmony-arm64-0.25.12.tgz",
      "integrity": "sha512-rm0YWsqUSRrjncSXGA7Zv78Nbnw4XL6/dzr20cyrQf7ZmRcsovpcRBdhD43Nuk3y7XIoW2OxMVvwuRvk9XdASg==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "openharmony"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/sunos-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/sunos-x64/-/sunos-x64-0.25.12.tgz",
      "integrity": "sha512-3wGSCDyuTHQUzt0nV7bocDy72r2lI33QL3gkDNGkod22EsYl04sMf0qLb8luNKTOmgF/eDEDP5BFNwoBKH441w==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "sunos"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/win32-arm64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/win32-arm64/-/win32-arm64-0.25.12.tgz",
      "integrity": "sha512-rMmLrur64A7+DKlnSuwqUdRKyd3UE7oPJZmnljqEptesKM8wx9J8gx5u0+9Pq0fQQW8vqeKebwNXdfOyP+8Bsg==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/win32-ia32": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/win32-ia32/-/win32-ia32-0.25.12.tgz",
      "integrity": "sha512-HkqnmmBoCbCwxUKKNPBixiWDGCpQGVsrQfJoVGYLPT41XWF8lHuE5N6WhVia2n4o5QK5M4tYr21827fNhi4byQ==",
      "cpu": [
        "ia32"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@esbuild/win32-x64": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/@esbuild/win32-x64/-/win32-x64-0.25.12.tgz",
      "integrity": "sha512-alJC0uCZpTFrSL0CCDjcgleBXPnCrEAhTBILpeAp7M/OFgoqtAetfBzX0xM00MUsVVPpVjlPuMbREqnZCXaTnA==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ],
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/@google/genai": {
      "version": "1.30.0",
      "resolved": "https://registry.npmjs.org/@google/genai/-/genai-1.30.0.tgz",
      "integrity": "sha512-3MRcgczBFbUat1wIlZoLJ0vCCfXgm7Qxjh59cZi2X08RgWLtm9hKOspzp7TOg1TV2e26/MLxR2GR5yD5GmBV2w==",
      "license": "Apache-2.0",
      "dependencies": {
        "google-auth-library": "^10.3.0",
        "ws": "^8.18.0"
      },
      "engines": {
        "node": ">=20.0.0"
      },
      "peerDependencies": {
        "@modelcontextprotocol/sdk": "^1.20.1"
      },
      "peerDependenciesMeta": {
        "@modelcontextprotocol/sdk": {
          "optional": true
        }
      }
    },
    "node_modules/@isaacs/cliui": {
      "version": "8.0.2",
      "resolved": "https://registry.npmjs.org/@isaacs/cliui/-/cliui-8.0.2.tgz",
      "integrity": "sha512-O8jcjabXaleOG9DQ0+ARXWZBTfnP4WNAqzuiJK7ll44AmxGKv/J2M4TPjxjY3znBCfvBXFzucm1twdyFybFqEA==",
      "license": "ISC",
      "dependencies": {
        "string-width": "^5.1.2",
        "string-width-cjs": "npm:string-width@^4.2.0",
        "strip-ansi": "^7.0.1",
        "strip-ansi-cjs": "npm:strip-ansi@^6.0.1",
        "wrap-ansi": "^8.1.0",
        "wrap-ansi-cjs": "npm:wrap-ansi@^7.0.0"
      },
      "engines": {
        "node": ">=12"
      }
    },
    "node_modules/@jridgewell/gen-mapping": {
      "version": "0.3.13",
      "resolved": "https://registry.npmjs.org/@jridgewell/gen-mapping/-/gen-mapping-0.3.13.tgz",
      "integrity": "sha512-2kkt/7niJ6MgEPxF0bYdQ6etZaA+fQvDcLKckhy1yIQOzaoKjBBjSj63/aLVjYE3qhRt5dvM+uUyfCg6UKCBbA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@jridgewell/sourcemap-codec": "^1.5.0",
        "@jridgewell/trace-mapping": "^0.3.24"
      }
    },
    "node_modules/@jridgewell/remapping": {
      "version": "2.3.5",
      "resolved": "https://registry.npmjs.org/@jridgewell/remapping/-/remapping-2.3.5.tgz",
      "integrity": "sha512-LI9u/+laYG4Ds1TDKSJW2YPrIlcVYOwi2fUC6xB43lueCjgxV4lffOCZCtYFiH6TNOX+tQKXx97T4IKHbhyHEQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@jridgewell/gen-mapping": "^0.3.5",
        "@jridgewell/trace-mapping": "^0.3.24"
      }
    },
    "node_modules/@jridgewell/resolve-uri": {
      "version": "3.1.2",
      "resolved": "https://registry.npmjs.org/@jridgewell/resolve-uri/-/resolve-uri-3.1.2.tgz",
      "integrity": "sha512-bRISgCIjP20/tbWSPWMEi54QVPRZExkuD9lJL+UIxUKtwVJA8wW1Trb1jMs1RFXo1CBTNZ/5hpC9QvmKWdopKw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.0.0"
      }
    },
    "node_modules/@jridgewell/sourcemap-codec": {
      "version": "1.5.5",
      "resolved": "https://registry.npmjs.org/@jridgewell/sourcemap-codec/-/sourcemap-codec-1.5.5.tgz",
      "integrity": "sha512-cYQ9310grqxueWbl+WuIUIaiUaDcj7WOq5fVhEljNVgRfOUhY9fy2zTvfoqWsnebh8Sl70VScFbICvJnLKB0Og==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@jridgewell/trace-mapping": {
      "version": "0.3.31",
      "resolved": "https://registry.npmjs.org/@jridgewell/trace-mapping/-/trace-mapping-0.3.31.tgz",
      "integrity": "sha512-zzNR+SdQSDJzc8joaeP8QQoCQr8NuYx2dIIytl1QeBEZHJ9uW6hebsrYgbz8hJwUQao3TWCMtmfV8Nu1twOLAw==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@jridgewell/resolve-uri": "^3.1.0",
        "@jridgewell/sourcemap-codec": "^1.4.14"
      }
    },
    "node_modules/@pkgjs/parseargs": {
      "version": "0.11.0",
      "resolved": "https://registry.npmjs.org/@pkgjs/parseargs/-/parseargs-0.11.0.tgz",
      "integrity": "sha512-+1VkjdD0QBLPodGrJUeqarH8VAIvQODIbwh9XpP5Syisf7YoQgsJKPNFoqqLQlu+VQ/tVSshMR6loPMn8U+dPg==",
      "license": "MIT",
      "optional": true,
      "engines": {
        "node": ">=14"
      }
    },
    "node_modules/@rolldown/pluginutils": {
      "version": "1.0.0-beta.47",
      "resolved": "https://registry.npmjs.org/@rolldown/pluginutils/-/pluginutils-1.0.0-beta.47.tgz",
      "integrity": "sha512-8QagwMH3kNCuzD8EWL8R2YPW5e4OrHNSAHRFDdmFqEwEaD/KcNKjVoumo+gP2vW5eKB2UPbM6vTYiGZX0ixLnw==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@rollup/rollup-android-arm-eabi": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-android-arm-eabi/-/rollup-android-arm-eabi-4.53.2.tgz",
      "integrity": "sha512-yDPzwsgiFO26RJA4nZo8I+xqzh7sJTZIWQOxn+/XOdPE31lAvLIYCKqjV+lNH/vxE2L2iH3plKxDCRK6i+CwhA==",
      "cpu": [
        "arm"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "android"
      ]
    },
    "node_modules/@rollup/rollup-android-arm64": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-android-arm64/-/rollup-android-arm64-4.53.2.tgz",
      "integrity": "sha512-k8FontTxIE7b0/OGKeSN5B6j25EuppBcWM33Z19JoVT7UTXFSo3D9CdU39wGTeb29NO3XxpMNauh09B+Ibw+9g==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "android"
      ]
    },
    "node_modules/@rollup/rollup-darwin-arm64": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-darwin-arm64/-/rollup-darwin-arm64-4.53.2.tgz",
      "integrity": "sha512-A6s4gJpomNBtJ2yioj8bflM2oogDwzUiMl2yNJ2v9E7++sHrSrsQ29fOfn5DM/iCzpWcebNYEdXpaK4tr2RhfQ==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "darwin"
      ]
    },
    "node_modules/@rollup/rollup-darwin-x64": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-darwin-x64/-/rollup-darwin-x64-4.53.2.tgz",
      "integrity": "sha512-e6XqVmXlHrBlG56obu9gDRPW3O3hLxpwHpLsBJvuI8qqnsrtSZ9ERoWUXtPOkY8c78WghyPHZdmPhHLWNdAGEw==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "darwin"
      ]
    },
    "node_modules/@rollup/rollup-freebsd-arm64": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-freebsd-arm64/-/rollup-freebsd-arm64-4.53.2.tgz",
      "integrity": "sha512-v0E9lJW8VsrwPux5Qe5CwmH/CF/2mQs6xU1MF3nmUxmZUCHazCjLgYvToOk+YuuUqLQBio1qkkREhxhc656ViA==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "freebsd"
      ]
    },
    "node_modules/@rollup/rollup-freebsd-x64": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-freebsd-x64/-/rollup-freebsd-x64-4.53.2.tgz",
      "integrity": "sha512-ClAmAPx3ZCHtp6ysl4XEhWU69GUB1D+s7G9YjHGhIGCSrsg00nEGRRZHmINYxkdoJehde8VIsDC5t9C0gb6yqA==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "freebsd"
      ]
    },
    "node_modules/@rollup/rollup-linux-arm-gnueabihf": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm-gnueabihf/-/rollup-linux-arm-gnueabihf-4.53.2.tg
      "integrity": "sha512-EPlb95nUsz6Dd9Qy13fI5kUPXNSljaG9FiJ4YUGU1O/Q77i5DYFW5KR8g1OzTcdZUqQQ1KdDqsTohdFVwCwjqg==",
      "cpu": [
        "arm"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-arm-musleabihf": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm-musleabihf/-/rollup-linux-arm-musleabihf-4.53.2.
      "integrity": "sha512-BOmnVW+khAUX+YZvNfa0tGTEMVVEerOxN0pDk2E6N6DsEIa2Ctj48FOMfNDdrwinocKaC7YXUZ1pHlKpnkja/Q==",
      "cpu": [
        "arm"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-arm64-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm64-gnu/-/rollup-linux-arm64-gnu-4.53.2.tgz",
      "integrity": "sha512-Xt2byDZ+6OVNuREgBXr4+CZDJtrVso5woFtpKdGPhpTPHcNG7D8YXeQzpNbFRxzTVqJf7kvPMCub/pcGUWgBjA==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-arm64-musl": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-arm64-musl/-/rollup-linux-arm64-musl-4.53.2.tgz",
      "integrity": "sha512-+LdZSldy/I9N8+klim/Y1HsKbJ3BbInHav5qE9Iy77dtHC/pibw1SR/fXlWyAk0ThnpRKoODwnAuSjqxFRDHUQ==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-loong64-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-loong64-gnu/-/rollup-linux-loong64-gnu-4.53.2.tgz",
      "integrity": "sha512-8ms8sjmyc1jWJS6WdNSA23rEfdjWB30LH8Wqj0Cqvv7qSHnvw6kgMMXRdop6hkmGPlyYBdRPkjJnj3KCUHV/uQ==",
      "cpu": [
        "loong64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-ppc64-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-ppc64-gnu/-/rollup-linux-ppc64-gnu-4.53.2.tgz",
      "integrity": "sha512-3HRQLUQbpBDMmzoxPJYd3W6vrVHOo2cVW8RUo87Xz0JPJcBLBr5kZ1pGcQAhdZgX9VV7NbGNipah1omKKe23/g==",
      "cpu": [
        "ppc64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-riscv64-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-riscv64-gnu/-/rollup-linux-riscv64-gnu-4.53.2.tgz",
      "integrity": "sha512-fMjKi+ojnmIvhk34gZP94vjogXNNUKMEYs+EDaB/5TG/wUkoeua7p7VCHnE6T2Tx+iaghAqQX8teQzcvrYpaQA==",
      "cpu": [
        "riscv64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-riscv64-musl": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-riscv64-musl/-/rollup-linux-riscv64-musl-4.53.2.tgz"
      "integrity": "sha512-XuGFGU+VwUUV5kLvoAdi0Wz5Xbh2SrjIxCtZj6Wq8MDp4bflb/+ThZsVxokM7n0pcbkEr2h5/pzqzDYI7cCgLQ==",
      "cpu": [
        "riscv64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-s390x-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-s390x-gnu/-/rollup-linux-s390x-gnu-4.53.2.tgz",
      "integrity": "sha512-w6yjZF0P+NGzWR3AXWX9zc0DNEGdtvykB03uhonSHMRa+oWA6novflo2WaJr6JZakG2ucsyb+rvhrKac6NIy+w==",
      "cpu": [
        "s390x"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-x64-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-x64-gnu/-/rollup-linux-x64-gnu-4.53.2.tgz",
      "integrity": "sha512-yo8d6tdfdeBArzC7T/PnHd7OypfI9cbuZzPnzLJIyKYFhAQ8SvlkKtKBMbXDxe1h03Rcr7u++nFS7tqXz87Gtw==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-linux-x64-musl": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-linux-x64-musl/-/rollup-linux-x64-musl-4.53.2.tgz",
      "integrity": "sha512-ah59c1YkCxKExPP8O9PwOvs+XRLKwh/mV+3YdKqQ5AMQ0r4M4ZDuOrpWkUaqO7fzAHdINzV9tEVu8vNw48z0lA==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "linux"
      ]
    },
    "node_modules/@rollup/rollup-openharmony-arm64": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-openharmony-arm64/-/rollup-openharmony-arm64-4.53.2.tgz",
      "integrity": "sha512-4VEd19Wmhr+Zy7hbUsFZ6YXEiP48hE//KPLCSVNY5RMGX2/7HZ+QkN55a3atM1C/BZCGIgqN+xrVgtdak2S9+A==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "openharmony"
      ]
    },
    "node_modules/@rollup/rollup-win32-arm64-msvc": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-win32-arm64-msvc/-/rollup-win32-arm64-msvc-4.53.2.tgz",
      "integrity": "sha512-IlbHFYc/pQCgew/d5fslcy1KEaYVCJ44G8pajugd8VoOEI8ODhtb/j8XMhLpwHCMB3yk2J07ctup10gpw2nyMA==",
      "cpu": [
        "arm64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ]
    },
    "node_modules/@rollup/rollup-win32-ia32-msvc": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-win32-ia32-msvc/-/rollup-win32-ia32-msvc-4.53.2.tgz",
      "integrity": "sha512-lNlPEGgdUfSzdCWU176ku/dQRnA7W+Gp8d+cWv73jYrb8uT7HTVVxq62DUYxjbaByuf1Yk0RIIAbDzp+CnOTFg==",
      "cpu": [
        "ia32"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ]
    },
    "node_modules/@rollup/rollup-win32-x64-gnu": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-win32-x64-gnu/-/rollup-win32-x64-gnu-4.53.2.tgz",
      "integrity": "sha512-S6YojNVrHybQis2lYov1sd+uj7K0Q05NxHcGktuMMdIQ2VixGwAfbJ23NnlvvVV1bdpR2m5MsNBViHJKcA4ADw==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ]
    },
    "node_modules/@rollup/rollup-win32-x64-msvc": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/@rollup/rollup-win32-x64-msvc/-/rollup-win32-x64-msvc-4.53.2.tgz",
      "integrity": "sha512-k+/Rkcyx//P6fetPoLMb8pBeqJBNGx81uuf7iljX9++yNBVRDQgD04L+SVXmXmh5ZP4/WOp4mWF0kmi06PW2tA==",
      "cpu": [
        "x64"
      ],
      "dev": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "win32"
      ]
    },
    "node_modules/@types/babel__core": {
      "version": "7.20.5",
      "resolved": "https://registry.npmjs.org/@types/babel__core/-/babel__core-7.20.5.tgz",
      "integrity": "sha512-qoQprZvz5wQFJwMDqeseRXWv3rqMvhgpbXFfVyWhbx9X47POIA6i/+dXefEmZKoAgOaTdaIgNSMqMIU61yRyzA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/parser": "^7.20.7",
        "@babel/types": "^7.20.7",
        "@types/babel__generator": "*",
        "@types/babel__template": "*",
        "@types/babel__traverse": "*"
      }
    },
    "node_modules/@types/babel__generator": {
      "version": "7.27.0",
      "resolved": "https://registry.npmjs.org/@types/babel__generator/-/babel__generator-7.27.0.tgz",
      "integrity": "sha512-ufFd2Xi92OAVPYsy+P4n7/U7e68fex0+Ee8gSG9KX7eo084CWiQ4sdxktvdl0bOPupXtVJPY19zk6EwWqUQ8lg==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/types": "^7.0.0"
      }
    },
    "node_modules/@types/babel__template": {
      "version": "7.4.4",
      "resolved": "https://registry.npmjs.org/@types/babel__template/-/babel__template-7.4.4.tgz",
      "integrity": "sha512-h/NUaSyG5EyxBIp8YRxo4RMe2/qQgvyowRwVMzhYhBCONbW8PUsg4lkFMrhgZhUe5z3L3MiLDuvyJ/CaPa2A8A==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/parser": "^7.1.0",
        "@babel/types": "^7.0.0"
      }
    },
    "node_modules/@types/babel__traverse": {
      "version": "7.28.0",
      "resolved": "https://registry.npmjs.org/@types/babel__traverse/-/babel__traverse-7.28.0.tgz",
      "integrity": "sha512-8PvcXf70gTDZBgt9ptxJ8elBeBjcLOAcOtoO/mPJjtji1+CdGbHgm77om1GrsPxsiE+uXIpNSK64UYaIwQXd4Q==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/types": "^7.28.2"
      }
    },
    "node_modules/@types/debug": {
      "version": "4.1.12",
      "resolved": "https://registry.npmjs.org/@types/debug/-/debug-4.1.12.tgz",
      "integrity": "sha512-vIChWdVG3LG1SMxEvI/AK+FWJthlrqlTu7fbrlywTkkaONwk/UAGaULXRlf8vkzFBLVm0zkMdCquhL5aOjhXPQ==",
      "license": "MIT",
      "dependencies": {
        "@types/ms": "*"
      }
    },
    "node_modules/@types/estree": {
      "version": "1.0.8",
      "resolved": "https://registry.npmjs.org/@types/estree/-/estree-1.0.8.tgz",
      "integrity": "sha512-dWHzHa2WqEXI/O1E9OjrocMTKJl2mSrEolh1Iomrv6U+JuNwaHXsXx9bLu5gG7BUWFIN0skIQJQ/L1rIex4X6w==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/@types/hast": {
      "version": "2.3.10",
      "resolved": "https://registry.npmjs.org/@types/hast/-/hast-2.3.10.tgz",
      "integrity": "sha512-McWspRw8xx8J9HurkVBfYj0xKoE25tOFlHGdx4MJ5xORQrMGZNqJhVQWaIbm6Oyla5kYOXtDiopzKRJzEOkwJw==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2"
      }
    },
    "node_modules/@types/mdast": {
      "version": "3.0.15",
      "resolved": "https://registry.npmjs.org/@types/mdast/-/mdast-3.0.15.tgz",
      "integrity": "sha512-LnwD+mUEfxWMa1QpDraczIn6k0Ee3SMicuYSSzS6ZYl2gKS09EClnJYGd8Du6rfc5r/GZEk5o1mRb8TaTj03sQ==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2"
      }
    },
    "node_modules/@types/ms": {
      "version": "2.1.0",
      "resolved": "https://registry.npmjs.org/@types/ms/-/ms-2.1.0.tgz",
      "integrity": "sha512-GsCCIZDE/p3i96vtEqx+7dBUGXrc7zeSK3wwPHIaRThS+9OhWIXRqzs4d6k1SVU8g91DrNRWxWUGhp5KXQb2VA==",
      "license": "MIT"
    },
    "node_modules/@types/node": {
      "version": "22.19.1",
      "resolved": "https://registry.npmjs.org/@types/node/-/node-22.19.1.tgz",
      "integrity": "sha512-LCCV0HdSZZZb34qifBsyWlUmok6W7ouER+oQIGBScS8EsZsQbrtFTUrDX4hOl+CS6p7cnNC4td+qrSVGSCTUfQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "undici-types": "~6.21.0"
      }
    },
    "node_modules/@types/react": {
      "version": "19.2.6",
      "resolved": "https://registry.npmjs.org/@types/react/-/react-19.2.6.tgz",
      "integrity": "sha512-p/jUvulfgU7oKtj6Xpk8cA2Y1xKTtICGpJYeJXz2YVO2UcvjQgeRMLDGfDeqeRW2Ta+0QNFwcc8X3GH8SxZz6w==",
      "license": "MIT",
      "dependencies": {
        "csstype": "^3.2.2"
      }
    },
    "node_modules/@types/unist": {
      "version": "2.0.11",
      "resolved": "https://registry.npmjs.org/@types/unist/-/unist-2.0.11.tgz",
      "integrity": "sha512-CmBKiL6NNo/OqgmMn95Fk9Whlp2mtvIv+KNpQKN2F4SjvrEesubTRWGYSg+BnWZOnlCaSTU1sMpsBOzgbYhnsA==",
      "license": "MIT"
    },
    "node_modules/@vitejs/plugin-react": {
      "version": "5.1.1",
      "resolved": "https://registry.npmjs.org/@vitejs/plugin-react/-/plugin-react-5.1.1.tgz",
      "integrity": "sha512-WQfkSw0QbQ5aJ2CHYw23ZGkqnRwqKHD/KYsMeTkZzPT4Jcf0DcBxBtwMJxnu6E7oxw5+JC6ZAiePgh28uJ1HBA==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@babel/core": "^7.28.5",
        "@babel/plugin-transform-react-jsx-self": "^7.27.1",
        "@babel/plugin-transform-react-jsx-source": "^7.27.1",
        "@rolldown/pluginutils": "1.0.0-beta.47",
        "@types/babel__core": "^7.20.5",
        "react-refresh": "^0.18.0"
      },
      "engines": {
        "node": "^20.19.0 || >=22.12.0"
      },
      "peerDependencies": {
        "vite": "^4.2.0 || ^5.0.0 || ^6.0.0 || ^7.0.0"
      }
    },
    "node_modules/agent-base": {
      "version": "7.1.4",
      "resolved": "https://registry.npmjs.org/agent-base/-/agent-base-7.1.4.tgz",
      "integrity": "sha512-MnA+YT8fwfJPgBx3m60MNqakm30XOkyIoH1y6huTQvC0PwZG7ki8NacLBcrPbNoo8vEZy7Jpuk7+jMO+CUovTQ==",
      "license": "MIT",
      "engines": {
        "node": ">= 14"
      }
    },
    "node_modules/ansi-regex": {
      "version": "6.2.2",
      "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-6.2.2.tgz",
      "integrity": "sha512-Bq3SmSpyFHaWjPk8If9yc6svM8c56dB5BAtW4Qbw5jHTwwXXcTLoRMkpDJp6VL0XzlWaCHTXrkFURMYmD0sLqg==",
      "license": "MIT",
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/chalk/ansi-regex?sponsor=1"
      }
    },
    "node_modules/ansi-styles": {
      "version": "6.2.3",
      "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-6.2.3.tgz",
      "integrity": "sha512-4Dj6M28JB+oAH8kFkTLUo+a2jwOFkuqb3yucU0CANcRRUbxS0cP0nZYCGjcc3BNXwRIsUVmDGgzawme7zvJHvg==",
      "license": "MIT",
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/chalk/ansi-styles?sponsor=1"
      }
    },
    "node_modules/bail": {
      "version": "2.0.2",
      "resolved": "https://registry.npmjs.org/bail/-/bail-2.0.2.tgz",
      "integrity": "sha512-0xO6mYd7JB2YesxDKplafRpsiOzPt9V02ddPCLbY1xYGPOX24NTyN50qnUxgCPcSoYMhKpAuBTjQoRZCAkUDRw==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/balanced-match": {
      "version": "1.0.2",
      "resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",
      "integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw==",
      "license": "MIT"
    },
    "node_modules/base64-js": {
      "version": "1.5.1",
      "resolved": "https://registry.npmjs.org/base64-js/-/base64-js-1.5.1.tgz",
      "integrity": "sha512-AKpaYlHn8t4SVbOHCy+b5+KKgvR4vrsD8vbvrbiQJps7fKDTkjkDry6ji0rUJjC0kzbNePLwzxq8iypo41qeWA==",
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/feross"
        },
        {
          "type": "patreon",
          "url": "https://www.patreon.com/feross"
        },
        {
          "type": "consulting",
          "url": "https://feross.org/support"
        }
      ],
      "license": "MIT"
    },
    "node_modules/baseline-browser-mapping": {
      "version": "2.8.29",
      "resolved": "https://registry.npmjs.org/baseline-browser-mapping/-/baseline-browser-mapping-2.8.29.tgz",
      "integrity": "sha512-sXdt2elaVnhpDNRDz+1BDx1JQoJRuNk7oVlAlbGiFkLikHCAQiccexF/9e91zVi6RCgqspl04aP+6Cnl9zRLrA==",
      "dev": true,
      "license": "Apache-2.0",
      "bin": {
        "baseline-browser-mapping": "dist/cli.js"
      }
    },
    "node_modules/bignumber.js": {
      "version": "9.3.1",
      "resolved": "https://registry.npmjs.org/bignumber.js/-/bignumber.js-9.3.1.tgz",
      "integrity": "sha512-Ko0uX15oIUS7wJ3Rb30Fs6SkVbLmPBAKdlm7q9+ak9bbIeFf0MwuBsQV6z7+X768/cHsfg+WlysDWJcmthjsjQ==",
      "license": "MIT",
      "engines": {
        "node": "*"
      }
    },
    "node_modules/brace-expansion": {
      "version": "2.0.2",
      "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.2.tgz",
      "integrity": "sha512-Jt0vHyM+jmUBqojB7E1NIYadt0vI0Qxjxd2TErW94wDz+E2LAm5vKMXXwg6ZZBTHPuUlDgQHKXvjGBdfcF1ZDQ==",
      "license": "MIT",
      "dependencies": {
        "balanced-match": "^1.0.0"
      }
    },
    "node_modules/browserslist": {
      "version": "4.28.0",
      "resolved": "https://registry.npmjs.org/browserslist/-/browserslist-4.28.0.tgz",
      "integrity": "sha512-tbydkR/CxfMwelN0vwdP/pLkDwyAASZ+VfWm4EOwlB6SWhx1sYnWLqo8N5j0rAzPfzfRaxt0mM/4wPU/Su84RQ==",
      "dev": true,
      "funding": [
        {
          "type": "opencollective",
          "url": "https://opencollective.com/browserslist"
        },
        {
          "type": "tidelift",
          "url": "https://tidelift.com/funding/github/npm/browserslist"
        },
        {
          "type": "github",
          "url": "https://github.com/sponsors/ai"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "baseline-browser-mapping": "^2.8.25",
        "caniuse-lite": "^1.0.30001754",
        "electron-to-chromium": "^1.5.249",
        "node-releases": "^2.0.27",
        "update-browserslist-db": "^1.1.4"
      },
      "bin": {
        "browserslist": "cli.js"
      },
      "engines": {
        "node": "^6 || ^7 || ^8 || ^9 || ^10 || ^11 || ^12 || >=13.7"
      }
    },
    "node_modules/buffer-equal-constant-time": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/buffer-equal-constant-time/-/buffer-equal-constant-time-1.0.1.tgz",
      "integrity": "sha512-zRpUiDwd/xk6ADqPMATG8vc9VPrkck7T07OIx0gnjmJAnHnTVXNQG3vfvWNuiZIkwu9KrKdA1iJKfsfTVxE6NA==",
      "license": "BSD-3-Clause"
    },
    "node_modules/caniuse-lite": {
      "version": "1.0.30001755",
      "resolved": "https://registry.npmjs.org/caniuse-lite/-/caniuse-lite-1.0.30001755.tgz",
      "integrity": "sha512-44V+Jm6ctPj7R52Na4TLi3Zri4dWUljJd+RDm+j8LtNCc/ihLCT+X1TzoOAkRETEWqjuLnh9581Tl80FvK7jVA==",
      "dev": true,
      "funding": [
        {
          "type": "opencollective",
          "url": "https://opencollective.com/browserslist"
        },
        {
          "type": "tidelift",
          "url": "https://tidelift.com/funding/github/npm/caniuse-lite"
        },
        {
          "type": "github",
          "url": "https://github.com/sponsors/ai"
        }
      ],
      "license": "CC-BY-4.0"
    },
    "node_modules/character-entities": {
      "version": "2.0.2",
      "resolved": "https://registry.npmjs.org/character-entities/-/character-entities-2.0.2.tgz",
      "integrity": "sha512-shx7oQ0Awen/BRIdkjkvz54PnEEI/EjwXDSIZp86/KKdbafHh1Df/RYGBhn4hbe2+uKC9FnT5UCEdyPz3ai9hQ==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/color-convert": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",
      "integrity": "sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHCaElKQ==",
      "license": "MIT",
      "dependencies": {
        "color-name": "~1.1.4"
      },
      "engines": {
        "node": ">=7.0.0"
      }
    },
    "node_modules/color-name": {
      "version": "1.1.4",
      "resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",
      "integrity": "sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGXgAsQA==",
      "license": "MIT"
    },
    "node_modules/comma-separated-tokens": {
      "version": "2.0.3",
      "resolved": "https://registry.npmjs.org/comma-separated-tokens/-/comma-separated-tokens-2.0.3.tgz",
      "integrity": "sha512-Fu4hJdvzeylCfQPp9SGWidpzrMs7tTrlu6Vb8XGaRGck8QSNZJJp538Wrb60Lax4fPwR64ViY468OIUTbRlGZg==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/convert-source-map": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/convert-source-map/-/convert-source-map-2.0.0.tgz",
      "integrity": "sha512-Kvp459HrV2FEJ1CAsi1Ku+MY3kasH19TFykTz2xWmMeq6bk2NU3XXvfJ+Q61m0xktWwt+1HSYf3JZsTms3aRJg==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/cross-spawn": {
      "version": "7.0.6",
      "resolved": "https://registry.npmjs.org/cross-spawn/-/cross-spawn-7.0.6.tgz",
      "integrity": "sha512-uV2QOWP2nWzsy2aMp8aRibhi9dlzF5Hgh5SHaB9OiTGEyDTiJJyx0uy51QXdyWbtAHNua4XJzUKca3OzKUd3vA==",
      "license": "MIT",
      "dependencies": {
        "path-key": "^3.1.0",
        "shebang-command": "^2.0.0",
        "which": "^2.0.1"
      },
      "engines": {
        "node": ">= 8"
      }
    },
    "node_modules/csstype": {
      "version": "3.2.3",
      "resolved": "https://registry.npmjs.org/csstype/-/csstype-3.2.3.tgz",
      "integrity": "sha512-z1HGKcYy2xA8AGQfwrn0PAy+PB7X/GSj3UVJW9qKyn43xWa+gl5nXmU4qqLMRzWVLFC8KusUX8T/0kCiOYpAIQ==",
      "license": "MIT"
    },
    "node_modules/data-uri-to-buffer": {
      "version": "4.0.1",
      "resolved": "https://registry.npmjs.org/data-uri-to-buffer/-/data-uri-to-buffer-4.0.1.tgz",
      "integrity": "sha512-0R9ikRb668HB7QDxT1vkpuUBtqc53YyAwMwGeUFKRojY/NWKvdZ+9UYtRfGmhqNbRkTSVpMbmyhXipFFv2cb/A==",
      "license": "MIT",
      "engines": {
        "node": ">= 12"
      }
    },
    "node_modules/debug": {
      "version": "4.4.3",
      "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.3.tgz",
      "integrity": "sha512-RGwwWnwQvkVfavKVt22FGLw+xYSdzARwm0ru6DhTVA3umU5hZc28V3kO4stgYryrTlLpuvgI9GiijltAjNbcqA==",
      "license": "MIT",
      "dependencies": {
        "ms": "^2.1.3"
      },
      "engines": {
        "node": ">=6.0"
      },
      "peerDependenciesMeta": {
        "supports-color": {
          "optional": true
        }
      }
    },
    "node_modules/decode-named-character-reference": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/decode-named-character-reference/-/decode-named-character-reference-1.2.0
      "integrity": "sha512-c6fcElNV6ShtZXmsgNgFFV5tVX2PaV4g+MOAkb8eXHvn6sryJBrZa9r0zV6+dtTyoCKxtDy5tyQ5ZwQuidtd+Q==",
      "license": "MIT",
      "dependencies": {
        "character-entities": "^2.0.0"
      },
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/dequal": {
      "version": "2.0.3",
      "resolved": "https://registry.npmjs.org/dequal/-/dequal-2.0.3.tgz",
      "integrity": "sha512-0je+qPKHEMohvfRTCEo3CrPG6cAzAYgmzKyxRiYSSDkS6eGJdyVJm7WaYA5ECaAD9wLB2T4EEeymA5aFVcYXCA==",
      "license": "MIT",
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/diff": {
      "version": "5.2.0",
      "resolved": "https://registry.npmjs.org/diff/-/diff-5.2.0.tgz",
      "integrity": "sha512-uIFDxqpRZGZ6ThOk84hEfqWoHx2devRFvpTZcTHur85vImfaxUbTW9Ryh4CpCuDnToOP1CEtXKIgytHBPVff5A==",
      "license": "BSD-3-Clause",
      "engines": {
        "node": ">=0.3.1"
      }
    },
    "node_modules/eastasianwidth": {
      "version": "0.2.0",
      "resolved": "https://registry.npmjs.org/eastasianwidth/-/eastasianwidth-0.2.0.tgz",
      "integrity": "sha512-I88TYZWc9XiYHRQ4/3c5rjjfgkjhLyW2luGIheGERbNQ6OY7yTybanSpDXZa8y7VUP9YmDcYa+eyq4ca7iLqWA==",
      "license": "MIT"
    },
    "node_modules/ecdsa-sig-formatter": {
      "version": "1.0.11",
      "resolved": "https://registry.npmjs.org/ecdsa-sig-formatter/-/ecdsa-sig-formatter-1.0.11.tgz",
      "integrity": "sha512-nagl3RYrbNv6kQkeJIpt6NJZy8twLB/2vtz6yN9Z4vRKHN4/QZJIEbqohALSgwKdnksuY3k5Addp5lg8sVoVcQ==",
      "license": "Apache-2.0",
      "dependencies": {
        "safe-buffer": "^5.0.1"
      }
    },
    "node_modules/electron-to-chromium": {
      "version": "1.5.255",
      "resolved": "https://registry.npmjs.org/electron-to-chromium/-/electron-to-chromium-1.5.255.tgz",
      "integrity": "sha512-Z9oIp4HrFF/cZkDPMpz2XSuVpc1THDpT4dlmATFlJUIBVCy9Vap5/rIXsASP1CscBacBqhabwh8vLctqBwEerQ==",
      "dev": true,
      "license": "ISC"
    },
    "node_modules/emoji-regex": {
      "version": "9.2.2",
      "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-9.2.2.tgz",
      "integrity": "sha512-L18DaJsXSUk2+42pv8mLs5jJT2hqFkFE4j21wOmgbUqsZ2hL72NsUU785g9RXgo3s0ZNgVl42TiHp3ZtOv/Vyg==",
      "license": "MIT"
    },
    "node_modules/esbuild": {
      "version": "0.25.12",
      "resolved": "https://registry.npmjs.org/esbuild/-/esbuild-0.25.12.tgz",
      "integrity": "sha512-bbPBYYrtZbkt6Os6FiTLCTFxvq4tt3JKall1vRwshA3fdVztsLAatFaZobhkBC8/BrPetoa0oksYoKXoG4ryJg==",
      "dev": true,
      "hasInstallScript": true,
      "license": "MIT",
      "bin": {
        "esbuild": "bin/esbuild"
      },
      "engines": {
        "node": ">=18"
      },
      "optionalDependencies": {
        "@esbuild/aix-ppc64": "0.25.12",
        "@esbuild/android-arm": "0.25.12",
        "@esbuild/android-arm64": "0.25.12",
        "@esbuild/android-x64": "0.25.12",
        "@esbuild/darwin-arm64": "0.25.12",
        "@esbuild/darwin-x64": "0.25.12",
        "@esbuild/freebsd-arm64": "0.25.12",
        "@esbuild/freebsd-x64": "0.25.12",
        "@esbuild/linux-arm": "0.25.12",
        "@esbuild/linux-arm64": "0.25.12",
        "@esbuild/linux-ia32": "0.25.12",
        "@esbuild/linux-loong64": "0.25.12",
        "@esbuild/linux-mips64el": "0.25.12",
        "@esbuild/linux-ppc64": "0.25.12",
        "@esbuild/linux-riscv64": "0.25.12",
        "@esbuild/linux-s390x": "0.25.12",
        "@esbuild/linux-x64": "0.25.12",
        "@esbuild/netbsd-arm64": "0.25.12",
        "@esbuild/netbsd-x64": "0.25.12",
        "@esbuild/openbsd-arm64": "0.25.12",
        "@esbuild/openbsd-x64": "0.25.12",
        "@esbuild/openharmony-arm64": "0.25.12",
        "@esbuild/sunos-x64": "0.25.12",
        "@esbuild/win32-arm64": "0.25.12",
        "@esbuild/win32-ia32": "0.25.12",
        "@esbuild/win32-x64": "0.25.12"
      }
    },
    "node_modules/escalade": {
      "version": "3.2.0",
      "resolved": "https://registry.npmjs.org/escalade/-/escalade-3.2.0.tgz",
      "integrity": "sha512-WUj2qlxaQtO4g6Pq5c29GTcWGDyd8itL8zTlipgECz3JesAiiOKotd8JU6otB3PACgG6xkJUyVhboMS+bje/jA==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/extend": {
      "version": "3.0.2",
      "resolved": "https://registry.npmjs.org/extend/-/extend-3.0.2.tgz",
      "integrity": "sha512-fjquC59cD7CyW6urNXK0FBufkZcoiGG80wTuPujX590cB5Ttln20E2UB4S/WARVqhXffZl2LNgS+gQdPIIim/g==",
      "license": "MIT"
    },
    "node_modules/fdir": {
      "version": "6.5.0",
      "resolved": "https://registry.npmjs.org/fdir/-/fdir-6.5.0.tgz",
      "integrity": "sha512-tIbYtZbucOs0BRGqPJkshJUYdL+SDH7dVM8gjy+ERp3WAUjLEFJE+02kanyHtwjWOnwrKYBiwAmM0p4kLJAnXg==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=12.0.0"
      },
      "peerDependencies": {
        "picomatch": "^3 || ^4"
      },
      "peerDependenciesMeta": {
        "picomatch": {
          "optional": true
        }
      }
    },
    "node_modules/fetch-blob": {
      "version": "3.2.0",
      "resolved": "https://registry.npmjs.org/fetch-blob/-/fetch-blob-3.2.0.tgz",
      "integrity": "sha512-7yAQpD2UMJzLi1Dqv7qFYnPbaPx7ZfFK6PiIxQ4PfkGPyNyl2Ugx+a/umUonmKqjhM4DnfbMvdX6otXq83soQQ==",
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/jimmywarting"
        },
        {
          "type": "paypal",
          "url": "https://paypal.me/jimmywarting"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "node-domexception": "^1.0.0",
        "web-streams-polyfill": "^3.0.3"
      },
      "engines": {
        "node": "^12.20 || >= 14.13"
      }
    },
    "node_modules/foreground-child": {
      "version": "3.3.1",
      "resolved": "https://registry.npmjs.org/foreground-child/-/foreground-child-3.3.1.tgz",
      "integrity": "sha512-gIXjKqtFuWEgzFRJA9WCQeSJLZDjgJUOMCMzxtvFq/37KojM1BFGufqsCy0r4qSQmYLsZYMeyRqzIWOMup03sw==",
      "license": "ISC",
      "dependencies": {
        "cross-spawn": "^7.0.6",
        "signal-exit": "^4.0.1"
      },
      "engines": {
        "node": ">=14"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      }
    },
    "node_modules/formdata-polyfill": {
      "version": "4.0.10",
      "resolved": "https://registry.npmjs.org/formdata-polyfill/-/formdata-polyfill-4.0.10.tgz",
      "integrity": "sha512-buewHzMvYL29jdeQTVILecSaZKnt/RJWjoZCF5OW60Z67/GmSLBkOFM7qh1PI3zFNtJbaZL5eQu1vLfazOwj4g==",
      "license": "MIT",
      "dependencies": {
        "fetch-blob": "^3.1.2"
      },
      "engines": {
        "node": ">=12.20.0"
      }
    },
    "node_modules/fsevents": {
      "version": "2.3.3",
      "resolved": "https://registry.npmjs.org/fsevents/-/fsevents-2.3.3.tgz",
      "integrity": "sha512-5xoDfX+fL7faATnagmWPpbFtwh/R77WmMMqqHGS65C3vvB0YHrgF+B1YmZ3441tMj5n63k0212XNoJwzlhffQw==",
      "dev": true,
      "hasInstallScript": true,
      "license": "MIT",
      "optional": true,
      "os": [
        "darwin"
      ],
      "engines": {
        "node": "^8.16.0 || ^10.6.0 || >=11.0.0"
      }
    },
    "node_modules/gaxios": {
      "version": "7.1.3",
      "resolved": "https://registry.npmjs.org/gaxios/-/gaxios-7.1.3.tgz",
      "integrity": "sha512-YGGyuEdVIjqxkxVH1pUTMY/XtmmsApXrCVv5EU25iX6inEPbV+VakJfLealkBtJN69AQmh1eGOdCl9Sm1UP6XQ==",
      "license": "Apache-2.0",
      "dependencies": {
        "extend": "^3.0.2",
        "https-proxy-agent": "^7.0.1",
        "node-fetch": "^3.3.2",
        "rimraf": "^5.0.1"
      },
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/gcp-metadata": {
      "version": "8.1.2",
      "resolved": "https://registry.npmjs.org/gcp-metadata/-/gcp-metadata-8.1.2.tgz",
      "integrity": "sha512-zV/5HKTfCeKWnxG0Dmrw51hEWFGfcF2xiXqcA3+J90WDuP0SvoiSO5ORvcBsifmx/FoIjgQN3oNOGaQ5PhLFkg==",
      "license": "Apache-2.0",
      "dependencies": {
        "gaxios": "^7.0.0",
        "google-logging-utils": "^1.0.0",
        "json-bigint": "^1.0.0"
      },
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/gensync": {
      "version": "1.0.0-beta.2",
      "resolved": "https://registry.npmjs.org/gensync/-/gensync-1.0.0-beta.2.tgz",
      "integrity": "sha512-3hN7NaskYvMDLQY55gnW3NQ+mesEAepTqlg+VEbj7zzqEMBVNhzcGYYeqFo/TlYz6eQiFcp1HcsCZO+nGgS8zg==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=6.9.0"
      }
    },
    "node_modules/glob": {
      "version": "10.5.0",
      "resolved": "https://registry.npmjs.org/glob/-/glob-10.5.0.tgz",
      "integrity": "sha512-DfXN8DfhJ7NH3Oe7cFmu3NCu1wKbkReJ8TorzSAFbSKrlNaQSKfIzqYqVY8zlbs2NLBbWpRiU52GX2PbaBVNkg==",
      "license": "ISC",
      "dependencies": {
        "foreground-child": "^3.1.0",
        "jackspeak": "^3.1.2",
        "minimatch": "^9.0.4",
        "minipass": "^7.1.2",
        "package-json-from-dist": "^1.0.0",
        "path-scurry": "^1.11.1"
      },
      "bin": {
        "glob": "dist/esm/bin.mjs"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      }
    },
    "node_modules/google-auth-library": {
      "version": "10.5.0",
      "resolved": "https://registry.npmjs.org/google-auth-library/-/google-auth-library-10.5.0.tgz",
      "integrity": "sha512-7ABviyMOlX5hIVD60YOfHw4/CxOfBhyduaYB+wbFWCWoni4N7SLcV46hrVRktuBbZjFC9ONyqamZITN7q3n32w==",
      "license": "Apache-2.0",
      "dependencies": {
        "base64-js": "^1.3.0",
        "ecdsa-sig-formatter": "^1.0.11",
        "gaxios": "^7.0.0",
        "gcp-metadata": "^8.0.0",
        "google-logging-utils": "^1.0.0",
        "gtoken": "^8.0.0",
        "jws": "^4.0.0"
      },
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/google-logging-utils": {
      "version": "1.1.3",
      "resolved": "https://registry.npmjs.org/google-logging-utils/-/google-logging-utils-1.1.3.tgz",
      "integrity": "sha512-eAmLkjDjAFCVXg7A1unxHsLf961m6y17QFqXqAXGj/gVkKFrEICfStRfwUlGNfeCEjNRa32JEWOUTlYXPyyKvA==",
      "license": "Apache-2.0",
      "engines": {
        "node": ">=14"
      }
    },
    "node_modules/gtoken": {
      "version": "8.0.0",
      "resolved": "https://registry.npmjs.org/gtoken/-/gtoken-8.0.0.tgz",
      "integrity": "sha512-+CqsMbHPiSTdtSO14O51eMNlrp9N79gmeqmXeouJOhfucAedHw9noVe/n5uJk3tbKE6a+6ZCQg3RPhVhHByAIw==",
      "license": "MIT",
      "dependencies": {
        "gaxios": "^7.0.0",
        "jws": "^4.0.0"
      },
      "engines": {
        "node": ">=18"
      }
    },
    "node_modules/hast-util-whitespace": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/hast-util-whitespace/-/hast-util-whitespace-2.0.1.tgz",
      "integrity": "sha512-nAxA0v8+vXSBDt3AnRUNjyRIQ0rD+ntpbAp4LnPkumc5M9yUbSMa4XDU9Q6etY4f1Wp4bNgvc1yjiZtsTTrSng==",
      "license": "MIT",
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/https-proxy-agent": {
      "version": "7.0.6",
      "resolved": "https://registry.npmjs.org/https-proxy-agent/-/https-proxy-agent-7.0.6.tgz",
      "integrity": "sha512-vK9P5/iUfdl95AI+JVyUuIcVtd4ofvtrOr3HNtM2yxC9bnMbEdp3x01OhQNnjb8IJYi38VlTE3mBXwcfvywuSw==",
      "license": "MIT",
      "dependencies": {
        "agent-base": "^7.1.2",
        "debug": "4"
      },
      "engines": {
        "node": ">= 14"
      }
    },
    "node_modules/inline-style-parser": {
      "version": "0.1.1",
      "resolved": "https://registry.npmjs.org/inline-style-parser/-/inline-style-parser-0.1.1.tgz",
      "integrity": "sha512-7NXolsK4CAS5+xvdj5OMMbI962hU/wvwoxk+LWR9Ek9bVtyuuYScDN6eS0rUm6TxApFpw7CX1o4uJzcd4AyD3Q==",
      "license": "MIT"
    },
    "node_modules/is-buffer": {
      "version": "2.0.5",
      "resolved": "https://registry.npmjs.org/is-buffer/-/is-buffer-2.0.5.tgz",
      "integrity": "sha512-i2R6zNFDwgEHJyQUtJEk0XFi1i0dPFn/oqjK3/vPCcDeJvW5NQ83V8QbicfF1SupOaB0h8ntgBC2YiE7dfyctQ==",
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/feross"
        },
        {
          "type": "patreon",
          "url": "https://www.patreon.com/feross"
        },
        {
          "type": "consulting",
          "url": "https://feross.org/support"
        }
      ],
      "license": "MIT",
      "engines": {
        "node": ">=4"
      }
    },
    "node_modules/is-fullwidth-code-point": {
      "version": "3.0.0",
      "resolved": "https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",
      "integrity": "sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKclKdizYaJUuHA83RLjb7nSuGnddCHGv0hk+KY7BMAlsWeK4Ueg6EV6XQg==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/is-plain-obj": {
      "version": "4.1.0",
      "resolved": "https://registry.npmjs.org/is-plain-obj/-/is-plain-obj-4.1.0.tgz",
      "integrity": "sha512-+Pgi+vMuUNkJyExiMBt5IlFoMyKnr5zhJ4Uspz58WOhBF5QoIZkFyNHIbBAtHwzVAgk5RtndVNsDRN61/mmDqg==",
      "license": "MIT",
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
      }
    },
    "node_modules/isexe": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/isexe/-/isexe-2.0.0.tgz",
      "integrity": "sha512-RHxMLp9lnKHGHRng9QFhRCMbYAcVpn69smSGcq3f36xjgVVWThj4qqLbTLlq7Ssj8B+fIQ1EuCEGI2lKsyQeIw==",
      "license": "ISC"
    },
    "node_modules/jackspeak": {
      "version": "3.4.3",
      "resolved": "https://registry.npmjs.org/jackspeak/-/jackspeak-3.4.3.tgz",
      "integrity": "sha512-OGlZQpz2yfahA/Rd1Y8Cd9SIEsqvXkLVoSw/cgwhnhFMDbsQFeZYoJJ7bIZBS9BcamUW96asq/npPWugM+RQBw==",
      "license": "BlueOak-1.0.0",
      "dependencies": {
        "@isaacs/cliui": "^8.0.2"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      },
      "optionalDependencies": {
        "@pkgjs/parseargs": "^0.11.0"
      }
    },
    "node_modules/js-tokens": {
      "version": "4.0.0",
      "resolved": "https://registry.npmjs.org/js-tokens/-/js-tokens-4.0.0.tgz",
      "integrity": "sha512-RdJUflcE3cUzKiMqQgsCu06FPu9UdIJO0beYbPhHN4k6apgJtifcoCtT9bcxOpYBtpD2kCM6Sbzg4CausW/PKQ==",
      "license": "MIT"
    },
    "node_modules/jsesc": {
      "version": "3.1.0",
      "resolved": "https://registry.npmjs.org/jsesc/-/jsesc-3.1.0.tgz",
      "integrity": "sha512-/sM3dO2FOzXjKQhJuo0Q173wf2KOo8t4I8vHy6lF9poUp7bKT0/NHE8fPX23PwfhnykfqnC2xRxOnVw5XuGIaA==",
      "dev": true,
      "license": "MIT",
      "bin": {
        "jsesc": "bin/jsesc"
      },
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/json-bigint": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/json-bigint/-/json-bigint-1.0.0.tgz",
      "integrity": "sha512-SiPv/8VpZuWbvLSMtTDU8hEfrZWg/mH/nV/b4o0CYbSxu1UIQPLdwKOCIyLQX+VIPO5vrLX3i8qtqFyhdPSUSQ==",
      "license": "MIT",
      "dependencies": {
        "bignumber.js": "^9.0.0"
      }
    },
    "node_modules/json5": {
      "version": "2.2.3",
      "resolved": "https://registry.npmjs.org/json5/-/json5-2.2.3.tgz",
      "integrity": "sha512-XmOWe7eyHYH14cLdVPoyg+GOH3rYX++KpzrylJwSW98t3Nk+U8XOl8FWKOgwtzdb8lXGf6zYwDUzeHMWfxasyg==",
      "dev": true,
      "license": "MIT",
      "bin": {
        "json5": "lib/cli.js"
      },
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/jwa": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/jwa/-/jwa-2.0.1.tgz",
      "integrity": "sha512-hRF04fqJIP8Abbkq5NKGN0Bbr3JxlQ+qhZufXVr0DvujKy93ZCbXZMHDL4EOtodSbCWxOqR8MS1tXA5hwqCXDg==",
      "license": "MIT",
      "dependencies": {
        "buffer-equal-constant-time": "^1.0.1",
        "ecdsa-sig-formatter": "1.0.11",
        "safe-buffer": "^5.0.1"
      }
    },
    "node_modules/jws": {
      "version": "4.0.0",
      "resolved": "https://registry.npmjs.org/jws/-/jws-4.0.0.tgz",
      "integrity": "sha512-KDncfTmOZoOMTFG4mBlG0qUIOlc03fmzH+ru6RgYVZhPkyiy/92Owlt/8UEN+a4TXR1FQetfIpJE8ApdvdVxTg==",
      "license": "MIT",
      "dependencies": {
        "jwa": "^2.0.0",
        "safe-buffer": "^5.0.1"
      }
    },
    "node_modules/kleur": {
      "version": "4.1.5",
      "resolved": "https://registry.npmjs.org/kleur/-/kleur-4.1.5.tgz",
      "integrity": "sha512-o+NO+8WrRiQEE4/7nwRJhN1HWpVmJm511pBHUxPLtp0BUISzlBplORYSmTclCnJvQq2tKu/sgl3xVpkc7ZWuQQ==",
      "license": "MIT",
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/loose-envify": {
      "version": "1.4.0",
      "resolved": "https://registry.npmjs.org/loose-envify/-/loose-envify-1.4.0.tgz",
      "integrity": "sha512-lyuxPGr/Wfhrlem2CL/UcnUc1zcqKAImBDzukY7Y5F/yQiNdko6+fRLevlw1HgMySw7f611UIY408EtxRSoK3Q==",
      "license": "MIT",
      "dependencies": {
        "js-tokens": "^3.0.0 || ^4.0.0"
      },
      "bin": {
        "loose-envify": "cli.js"
      }
    },
    "node_modules/lru-cache": {
      "version": "5.1.1",
      "resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-5.1.1.tgz",
      "integrity": "sha512-KpNARQA3Iwv+jTA0utUVVbrh+Jlrr1Fv0e56GGzAFOXN7dk/FviaDW8LHmK52DlcH4WP2n6gI8vN1aesBFgo9w==",
      "dev": true,
      "license": "ISC",
      "dependencies": {
        "yallist": "^3.0.2"
      }
    },
    "node_modules/mdast-util-definitions": {
      "version": "5.1.2",
      "resolved": "https://registry.npmjs.org/mdast-util-definitions/-/mdast-util-definitions-5.1.2.tgz",
      "integrity": "sha512-8SVPMuHqlPME/z3gqVwWY4zVXn8lqKv/pAhC57FuJ40ImXyBpmO5ukh98zB2v7Blql2FiHjHv9LVztSIqjY+MA==",
      "license": "MIT",
      "dependencies": {
        "@types/mdast": "^3.0.0",
        "@types/unist": "^2.0.0",
        "unist-util-visit": "^4.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/mdast-util-from-markdown": {
      "version": "1.3.1",
      "resolved": "https://registry.npmjs.org/mdast-util-from-markdown/-/mdast-util-from-markdown-1.3.1.tgz",
      "integrity": "sha512-4xTO/M8c82qBcnQc1tgpNtubGUW/Y1tBQ1B0i5CtSoelOLKFYlElIr3bvgREYYO5iRqbMY1YuqZng0GVOI8Qww==",
      "license": "MIT",
      "dependencies": {
        "@types/mdast": "^3.0.0",
        "@types/unist": "^2.0.0",
        "decode-named-character-reference": "^1.0.0",
        "mdast-util-to-string": "^3.1.0",
        "micromark": "^3.0.0",
        "micromark-util-decode-numeric-character-reference": "^1.0.0",
        "micromark-util-decode-string": "^1.0.0",
        "micromark-util-normalize-identifier": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0",
        "unist-util-stringify-position": "^3.0.0",
        "uvu": "^0.5.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/mdast-util-to-hast": {
      "version": "12.3.0",
      "resolved": "https://registry.npmjs.org/mdast-util-to-hast/-/mdast-util-to-hast-12.3.0.tgz",
      "integrity": "sha512-pits93r8PhnIoU4Vy9bjW39M2jJ6/tdHyja9rrot9uujkN7UTU9SDnE6WNJz/IGyQk3XHX6yNNtrBH6cQzm8Hw==",
      "license": "MIT",
      "dependencies": {
        "@types/hast": "^2.0.0",
        "@types/mdast": "^3.0.0",
        "mdast-util-definitions": "^5.0.0",
        "micromark-util-sanitize-uri": "^1.1.0",
        "trim-lines": "^3.0.0",
        "unist-util-generated": "^2.0.0",
        "unist-util-position": "^4.0.0",
        "unist-util-visit": "^4.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/mdast-util-to-string": {
      "version": "3.2.0",
      "resolved": "https://registry.npmjs.org/mdast-util-to-string/-/mdast-util-to-string-3.2.0.tgz",
      "integrity": "sha512-V4Zn/ncyN1QNSqSBxTrMOLpjr+IKdHl2v3KVLoWmDPscP4r9GcCi71gjgvUV1SFSKh92AjAG4peFuBl2/YgCJg==",
      "license": "MIT",
      "dependencies": {
        "@types/mdast": "^3.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/micromark": {
      "version": "3.2.0",
      "resolved": "https://registry.npmjs.org/micromark/-/micromark-3.2.0.tgz",
      "integrity": "sha512-uD66tJj54JLYq0De10AhWycZWGQNUvDI55xPgk2sQM5kn1JYlhbCMTtEeT27+vAhW2FBQxLlOmS3pmA7/2z4aA==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "@types/debug": "^4.0.0",
        "debug": "^4.0.0",
        "decode-named-character-reference": "^1.0.0",
        "micromark-core-commonmark": "^1.0.1",
        "micromark-factory-space": "^1.0.0",
        "micromark-util-character": "^1.0.0",
        "micromark-util-chunked": "^1.0.0",
        "micromark-util-combine-extensions": "^1.0.0",
        "micromark-util-decode-numeric-character-reference": "^1.0.0",
        "micromark-util-encode": "^1.0.0",
        "micromark-util-normalize-identifier": "^1.0.0",
        "micromark-util-resolve-all": "^1.0.0",
        "micromark-util-sanitize-uri": "^1.0.0",
        "micromark-util-subtokenize": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.1",
        "uvu": "^0.5.0"
      }
    },
    "node_modules/micromark-core-commonmark": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-core-commonmark/-/micromark-core-commonmark-1.1.0.tgz",
      "integrity": "sha512-BgHO1aRbolh2hcrzL2d1La37V0Aoz73ymF8rAcKnohLy93titmv62E0gP8Hrx9PKcKrqCZ1BbLGbP3bEhoXYlw==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "decode-named-character-reference": "^1.0.0",
        "micromark-factory-destination": "^1.0.0",
        "micromark-factory-label": "^1.0.0",
        "micromark-factory-space": "^1.0.0",
        "micromark-factory-title": "^1.0.0",
        "micromark-factory-whitespace": "^1.0.0",
        "micromark-util-character": "^1.0.0",
        "micromark-util-chunked": "^1.0.0",
        "micromark-util-classify-character": "^1.0.0",
        "micromark-util-html-tag-name": "^1.0.0",
        "micromark-util-normalize-identifier": "^1.0.0",
        "micromark-util-resolve-all": "^1.0.0",
        "micromark-util-subtokenize": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.1",
        "uvu": "^0.5.0"
      }
    },
    "node_modules/micromark-factory-destination": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-factory-destination/-/micromark-factory-destination-1.1.0.tgz",
      "integrity": "sha512-XaNDROBgx9SgSChd69pjiGKbV+nfHGDPVYFs5dOoDd7ZnMAE+Cuu91BCpsY8RT2NP9vo/B8pds2VQNCLiu0zhg==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-character": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-factory-label": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-factory-label/-/micromark-factory-label-1.1.0.tgz",
      "integrity": "sha512-OLtyez4vZo/1NjxGhcpDSbHQ+m0IIGnT8BoPamh+7jVlzLJBH98zzuCoUeMxvM6WsNeh8wx8cKvqLiPHEACn0w==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-character": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0",
        "uvu": "^0.5.0"
      }
    },
    "node_modules/micromark-factory-space": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-factory-space/-/micromark-factory-space-1.1.0.tgz",
      "integrity": "sha512-cRzEj7c0OL4Mw2v6nwzttyOZe8XY/Z8G0rzmWQZTBi/jjwyw/U4uqKtUORXQrR5bAZZnbTI/feRV/R7hc4jQYQ==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-character": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-factory-title": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-factory-title/-/micromark-factory-title-1.1.0.tgz",
      "integrity": "sha512-J7n9R3vMmgjDOCY8NPw55jiyaQnH5kBdV2/UXCtZIpnHH3P6nHUKaH7XXEYuWwx/xUJcawa8plLBEjMPU24HzQ==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-factory-space": "^1.0.0",
        "micromark-util-character": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-factory-whitespace": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-factory-whitespace/-/micromark-factory-whitespace-1.1.0.tgz",
      "integrity": "sha512-v2WlmiymVSp5oMg+1Q0N1Lxmt6pMhIHD457whWM7/GUlEks1hI9xj5w3zbc4uuMKXGisksZk8DzP2UyGbGqNsQ==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-factory-space": "^1.0.0",
        "micromark-util-character": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-util-character": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/micromark-util-character/-/micromark-util-character-1.2.0.tgz",
      "integrity": "sha512-lXraTwcX3yH/vMDaFWCQJP1uIszLVebzUa3ZHdrgxr7KEU/9mL4mVgCpGbyhvNLNlauROiNUq7WN5u7ndbY6xg==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-util-chunked": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-chunked/-/micromark-util-chunked-1.1.0.tgz",
      "integrity": "sha512-Ye01HXpkZPNcV6FiyoW2fGZDUw4Yc7vT0E9Sad83+bEDiCJ1uXu0S3mr8WLpsz3HaG3x2q0HM6CTuPdcZcluFQ==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-symbol": "^1.0.0"
      }
    },
    "node_modules/micromark-util-classify-character": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-classify-character/-/micromark-util-classify-character-1.1
      "integrity": "sha512-SL0wLxtKSnklKSUplok1WQFoGhUdWYKggKUiqhX+Swala+BtptGCu5iPRc+xvzJ4PXE/hwM3FNXsfEVgoZsWbw==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-character": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-util-combine-extensions": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-combine-extensions/-/micromark-util-combine-extensions-1.1
      "integrity": "sha512-Q20sp4mfNf9yEqDL50WwuWZHUrCO4fEyeDCnMGmG5Pr0Cz15Uo7KBs6jq+dq0EgX4DPwwrh9m0X+zPV1ypFvUA==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-chunked": "^1.0.0",
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-util-decode-numeric-character-reference": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-decode-numeric-character-reference/-/micromark-util-decode
      "integrity": "sha512-m9V0ExGv0jB1OT21mrWcuf4QhP46pH1KkfWy9ZEezqHKAxkj4mPCy3nIH1rkbdMlChLHX531eOrymlwyZIf2iw==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-symbol": "^1.0.0"
      }
    },
    "node_modules/micromark-util-decode-string": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-decode-string/-/micromark-util-decode-string-1.1.0.tgz",
      "integrity": "sha512-YphLGCK8gM1tG1bd54azwyrQRjCFcmgj2S2GoJDNnh4vYtnL38JS8M4gpxzOPNyHdNEpheyWXCTnnTDY3N+NVQ==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "decode-named-character-reference": "^1.0.0",
        "micromark-util-character": "^1.0.0",
        "micromark-util-decode-numeric-character-reference": "^1.0.0",
        "micromark-util-symbol": "^1.0.0"
      }
    },
    "node_modules/micromark-util-encode": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-encode/-/micromark-util-encode-1.1.0.tgz",
      "integrity": "sha512-EuEzTWSTAj9PA5GOAs992GzNh2dGQO52UvAbtSOMvXTxv3Criqb6IOzJUBCmEqrrXSblJIJBbFFv6zPxpreiJw==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT"
    },
    "node_modules/micromark-util-html-tag-name": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/micromark-util-html-tag-name/-/micromark-util-html-tag-name-1.2.0.tgz",
      "integrity": "sha512-VTQzcuQgFUD7yYztuQFKXT49KghjtETQ+Wv/zUjGSGBioZnkA4P1XXZPT1FHeJA6RwRXSF47yvJ1tsJdoxwO+Q==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT"
    },
    "node_modules/micromark-util-normalize-identifier": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-normalize-identifier/-/micromark-util-normalize-identifier
      "integrity": "sha512-N+w5vhqrBihhjdpM8+5Xsxy71QWqGn7HYNUvch71iV2PM7+E3uWGox1Qp90loa1ephtCxG2ftRV/Conitc6P2Q==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-symbol": "^1.0.0"
      }
    },
    "node_modules/micromark-util-resolve-all": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-resolve-all/-/micromark-util-resolve-all-1.1.0.tgz",
      "integrity": "sha512-b/G6BTMSg+bX+xVCshPTPyAu2tmA0E4X98NSR7eIbeC6ycCqCeE7wjfDIgzEbkzdEVJXRtOG4FbEm/uGbCRouA==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-types": "^1.0.0"
      }
    },
    "node_modules/micromark-util-sanitize-uri": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/micromark-util-sanitize-uri/-/micromark-util-sanitize-uri-1.2.0.tgz",
      "integrity": "sha512-QO4GXv0XZfWey4pYFndLUKEAktKkG5kZTdUNaTAkzbuJxn2tNBOr+QtxR2XpWaMhbImT2dPzyLrPXLlPhph34A==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-character": "^1.0.0",
        "micromark-util-encode": "^1.0.0",
        "micromark-util-symbol": "^1.0.0"
      }
    },
    "node_modules/micromark-util-subtokenize": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-subtokenize/-/micromark-util-subtokenize-1.1.0.tgz",
      "integrity": "sha512-kUQHyzRoxvZO2PuLzMt2P/dwVsTiivCK8icYTeR+3WgbuPqfHgPPy7nFKbeqRivBvn/3N3GBiNC+JRTMSxEC7A==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "micromark-util-chunked": "^1.0.0",
        "micromark-util-symbol": "^1.0.0",
        "micromark-util-types": "^1.0.0",
        "uvu": "^0.5.0"
      }
    },
    "node_modules/micromark-util-symbol": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-symbol/-/micromark-util-symbol-1.1.0.tgz",
      "integrity": "sha512-uEjpEYY6KMs1g7QfJ2eX1SQEV+ZT4rUD3UcF6l57acZvLNK7PBZL+ty82Z1qhK1/yXIY4bdx04FKMgR0g4IAag==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT"
    },
    "node_modules/micromark-util-types": {
      "version": "1.1.0",
      "resolved": "https://registry.npmjs.org/micromark-util-types/-/micromark-util-types-1.1.0.tgz",
      "integrity": "sha512-ukRBgie8TIAcacscVHSiddHjO4k/q3pnedmzMQ4iwDcK0FtFCohKOlFbaOL/mPgfnPsL3C1ZyxJa4sbWrBl3jg==",
      "funding": [
        {
          "type": "GitHub Sponsors",
          "url": "https://github.com/sponsors/unifiedjs"
        },
        {
          "type": "OpenCollective",
          "url": "https://opencollective.com/unified"
        }
      ],
      "license": "MIT"
    },
    "node_modules/minimatch": {
      "version": "9.0.5",
      "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-9.0.5.tgz",
      "integrity": "sha512-G6T0ZX48xgozx7587koeX9Ys2NYy6Gmv//P89sEte9V9whIapMNF4idKxnW2QtCcLiTWlb/wfCabAtAFWhhBow==",
      "license": "ISC",
      "dependencies": {
        "brace-expansion": "^2.0.1"
      },
      "engines": {
        "node": ">=16 || 14 >=14.17"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      }
    },
    "node_modules/minipass": {
      "version": "7.1.2",
      "resolved": "https://registry.npmjs.org/minipass/-/minipass-7.1.2.tgz",
      "integrity": "sha512-qOOzS1cBTWYF4BH8fVePDBOO9iptMnGUEZwNc/cMWnTV2nVLZ7VoNWEPHkYczZA0pdoA7dl6e7FL659nX9S2aw==",
      "license": "ISC",
      "engines": {
        "node": ">=16 || 14 >=14.17"
      }
    },
    "node_modules/mri": {
      "version": "1.2.0",
      "resolved": "https://registry.npmjs.org/mri/-/mri-1.2.0.tgz",
      "integrity": "sha512-tzzskb3bG8LvYGFF/mDTpq3jpI6Q9wc3LEmBaghu+DdCssd1FakN7Bc0hVNmEyGq1bq3RgfkCb3cmQLpNPOroA==",
      "license": "MIT",
      "engines": {
        "node": ">=4"
      }
    },
    "node_modules/ms": {
      "version": "2.1.3",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
      "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==",
      "license": "MIT"
    },
    "node_modules/nanoid": {
      "version": "3.3.11",
      "resolved": "https://registry.npmjs.org/nanoid/-/nanoid-3.3.11.tgz",
      "integrity": "sha512-N8SpfPUnUp1bK+PMYW8qSWdl9U+wwNWI4QKxOYDy9JAro3WMX7p2OeVRF9v+347pnakNevPmiHhNmZ2HbFA76w==",
      "dev": true,
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/ai"
        }
      ],
      "license": "MIT",
      "bin": {
        "nanoid": "bin/nanoid.cjs"
      },
      "engines": {
        "node": "^10 || ^12 || ^13.7 || ^14 || >=15.0.1"
      }
    },
    "node_modules/node-domexception": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/node-domexception/-/node-domexception-1.0.0.tgz",
      "integrity": "sha512-/jKZoMpw0F8GRwl4/eLROPA3cfcXtLApP0QzLmUT/HuPCZWyB7IY9ZrMeKw2O/nFIqPQB3PVM9aYm0F312AXDQ==",
      "deprecated": "Use your platform's native DOMException instead",
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/jimmywarting"
        },
        {
          "type": "github",
          "url": "https://paypal.me/jimmywarting"
        }
      ],
      "license": "MIT",
      "engines": {
        "node": ">=10.5.0"
      }
    },
    "node_modules/node-fetch": {
      "version": "3.3.2",
      "resolved": "https://registry.npmjs.org/node-fetch/-/node-fetch-3.3.2.tgz",
      "integrity": "sha512-dRB78srN/l6gqWulah9SrxeYnxeddIG30+GOqK/9OlLVyLg3HPnr6SqOWTWOXKRwC2eGYCkZ59NNuSgvSrpgOA==",
      "license": "MIT",
      "dependencies": {
        "data-uri-to-buffer": "^4.0.0",
        "fetch-blob": "^3.1.4",
        "formdata-polyfill": "^4.0.10"
      },
      "engines": {
        "node": "^12.20.0 || ^14.13.1 || >=16.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/node-fetch"
      }
    },
    "node_modules/node-releases": {
      "version": "2.0.27",
      "resolved": "https://registry.npmjs.org/node-releases/-/node-releases-2.0.27.tgz",
      "integrity": "sha512-nmh3lCkYZ3grZvqcCH+fjmQ7X+H0OeZgP40OierEaAptX4XofMh5kwNbWh7lBduUzCcV/8kZ+NDLCwm2iorIlA==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/object-assign": {
      "version": "4.1.1",
      "resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",
      "integrity": "sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==",
      "license": "MIT",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/package-json-from-dist": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/package-json-from-dist/-/package-json-from-dist-1.0.1.tgz",
      "integrity": "sha512-UEZIS3/by4OC8vL3P2dTXRETpebLI2NiI5vIrjaD/5UtrkFX/tNbwjTSRAGC/+7CAo2pIcBaRgWmcBBHcsaCIw==",
      "license": "BlueOak-1.0.0"
    },
    "node_modules/path-key": {
      "version": "3.1.1",
      "resolved": "https://registry.npmjs.org/path-key/-/path-key-3.1.1.tgz",
      "integrity": "sha512-ojmeN0qd+y0jszEtoY48r0Peq5dwMEkIlCOu6Q5f41lfkswXuKtYrhgoTpLnyIcHm24Uhqx+5Tqm2InSwLhE6Q==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/path-scurry": {
      "version": "1.11.1",
      "resolved": "https://registry.npmjs.org/path-scurry/-/path-scurry-1.11.1.tgz",
      "integrity": "sha512-Xa4Nw17FS9ApQFJ9umLiJS4orGjm7ZzwUrwamcGQuHSzDyth9boKDaycYdDcZDuqYATXw4HFXgaqWTctW/v1HA==",
      "license": "BlueOak-1.0.0",
      "dependencies": {
        "lru-cache": "^10.2.0",
        "minipass": "^5.0.0 || ^6.0.2 || ^7.0.0"
      },
      "engines": {
        "node": ">=16 || 14 >=14.18"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      }
    },
    "node_modules/path-scurry/node_modules/lru-cache": {
      "version": "10.4.3",
      "resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-10.4.3.tgz",
      "integrity": "sha512-JNAzZcXrCt42VGLuYz0zfAzDfAvJWW6AfYlDBQyDV5DClI2m5sAmK+OIO7s59XfsRsWHp02jAJrRadPRGTt6SQ==",
      "license": "ISC"
    },
    "node_modules/picocolors": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/picocolors/-/picocolors-1.1.1.tgz",
      "integrity": "sha512-xceH2snhtb5M9liqDsmEw56le376mTZkEX/jEb/RxNFyegNul7eNslCXP9FDj/Lcu0X8KEyMceP2ntpaHrDEVA==",
      "dev": true,
      "license": "ISC"
    },
    "node_modules/picomatch": {
      "version": "4.0.3",
      "resolved": "https://registry.npmjs.org/picomatch/-/picomatch-4.0.3.tgz",
      "integrity": "sha512-5gTmgEY/sqK6gFXLIsQNH19lWb4ebPDLA4SdLP7dsWkIXHWlG66oPuVvXSGFPppYZz8ZDZq0dYYrbHfBCVUb1Q==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/sponsors/jonschlinkert"
      }
    },
    "node_modules/postcss": {
      "version": "8.5.6",
      "resolved": "https://registry.npmjs.org/postcss/-/postcss-8.5.6.tgz",
      "integrity": "sha512-3Ybi1tAuwAP9s0r1UQ2J4n5Y0G05bJkpUIO0/bI9MhwmD70S5aTWbXGBwxHrelT+XM1k6dM0pk+SwNkpTRN7Pg==",
      "dev": true,
      "funding": [
        {
          "type": "opencollective",
          "url": "https://opencollective.com/postcss/"
        },
        {
          "type": "tidelift",
          "url": "https://tidelift.com/funding/github/npm/postcss"
        },
        {
          "type": "github",
          "url": "https://github.com/sponsors/ai"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "nanoid": "^3.3.11",
        "picocolors": "^1.1.1",
        "source-map-js": "^1.2.1"
      },
      "engines": {
        "node": "^10 || ^12 || >=14"
      }
    },
    "node_modules/prismjs": {
      "version": "1.30.0",
      "resolved": "https://registry.npmjs.org/prismjs/-/prismjs-1.30.0.tgz",
      "integrity": "sha512-DEvV2ZF2r2/63V+tK8hQvrR2ZGn10srHbXviTlcv7Kpzw8jWiNTqbVgjO3IY8RxrrOUF8VPMQQFysYYYv0YZxw==",
      "license": "MIT",
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/prop-types": {
      "version": "15.8.1",
      "resolved": "https://registry.npmjs.org/prop-types/-/prop-types-15.8.1.tgz",
      "integrity": "sha512-oj87CgZICdulUohogVAR7AjlC0327U4el4L6eAvOqCeudMDVU0NThNaV+b9Df4dXgSP1gXMTnPdhfe/2qDH5cg==",
      "license": "MIT",
      "dependencies": {
        "loose-envify": "^1.4.0",
        "object-assign": "^4.1.1",
        "react-is": "^16.13.1"
      }
    },
    "node_modules/prop-types/node_modules/react-is": {
      "version": "16.13.1",
      "resolved": "https://registry.npmjs.org/react-is/-/react-is-16.13.1.tgz",
      "integrity": "sha512-24e6ynE2H+OKt4kqsOvNd8kBpV65zoxbA4BVsEOB3ARVWQki/DHzaUoC5KuON/BiccDaCCTZBuOcfZs70kR8bQ==",
      "license": "MIT"
    },
    "node_modules/property-information": {
      "version": "6.5.0",
      "resolved": "https://registry.npmjs.org/property-information/-/property-information-6.5.0.tgz",
      "integrity": "sha512-PgTgs/BlvHxOu8QuEN7wi5A0OmXaBcHpmCSTehcs6Uuu9IkDIEo13Hy7n898RHfrQ49vKCoGeWZSaAK01nwVig==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/react": {
      "version": "19.2.0",
      "resolved": "https://registry.npmjs.org/react/-/react-19.2.0.tgz",
      "integrity": "sha512-tmbWg6W31tQLeB5cdIBOicJDJRR2KzXsV7uSK9iNfLWQ5bIZfxuPEHp7M8wiHyHnn0DD1i7w3Zmin0FtkrwoCQ==",
      "license": "MIT",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/react-dom": {
      "version": "19.2.0",
      "resolved": "https://registry.npmjs.org/react-dom/-/react-dom-19.2.0.tgz",
      "integrity": "sha512-UlbRu4cAiGaIewkPyiRGJk0imDN2T3JjieT6spoL2UeSf5od4n5LB/mQ4ejmxhCFT1tYe8IvaFulzynWovsEFQ==",
      "license": "MIT",
      "dependencies": {
        "scheduler": "^0.27.0"
      },
      "peerDependencies": {
        "react": "^19.2.0"
      }
    },
    "node_modules/react-is": {
      "version": "17.0.2",
      "resolved": "https://registry.npmjs.org/react-is/-/react-is-17.0.2.tgz",
      "integrity": "sha512-w2GsyukL62IJnlaff/nRegPQR94C/XXamvMWmSHRJ4y7Ts/4ocGRmTHvOs8PSE6pB3dWOrD/nueuU5sduBsQ4w==",
      "license": "MIT"
    },
    "node_modules/react-markdown": {
      "version": "8.0.0",
      "resolved": "https://registry.npmjs.org/react-markdown/-/react-markdown-8.0.0.tgz",
      "integrity": "sha512-qbrWpLny6Ef2xHqnYqtot948LXP+4FtC+MWIuaN1kvSnowM+r1qEeEHpSaU0TDBOisQuj+Qe6eFY15cNL3gLAw==",
      "license": "MIT",
      "dependencies": {
        "@types/hast": "^2.0.0",
        "@types/unist": "^2.0.0",
        "comma-separated-tokens": "^2.0.0",
        "hast-util-whitespace": "^2.0.0",
        "prop-types": "^15.0.0",
        "property-information": "^6.0.0",
        "react-is": "^17.0.0",
        "remark-parse": "^10.0.0",
        "remark-rehype": "^10.0.0",
        "space-separated-tokens": "^2.0.0",
        "style-to-object": "^0.3.0",
        "unified": "^10.0.0",
        "unist-util-visit": "^4.0.0",
        "vfile": "^5.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      },
      "peerDependencies": {
        "@types/react": ">=16",
        "react": ">=16"
      }
    },
    "node_modules/react-refresh": {
      "version": "0.18.0",
      "resolved": "https://registry.npmjs.org/react-refresh/-/react-refresh-0.18.0.tgz",
      "integrity": "sha512-QgT5//D3jfjJb6Gsjxv0Slpj23ip+HtOpnNgnb2S5zU3CB26G/IDPGoy4RJB42wzFE46DRsstbW6tKHoKbhAxw==",
      "dev": true,
      "license": "MIT",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/react-simple-code-editor": {
      "version": "0.13.1",
      "resolved": "https://registry.npmjs.org/react-simple-code-editor/-/react-simple-code-editor-0.13.1.tgz",
      "integrity": "sha512-XYeVwRZwgyKtjNIYcAEgg2FaQcCZwhbarnkJIV20U2wkCU9q/CPFBo8nRXrK4GXUz3AvbqZFsZRrpUTkqqEYyQ==",
      "license": "MIT",
      "peerDependencies": {
        "react": "*",
        "react-dom": "*"
      }
    },
    "node_modules/remark-parse": {
      "version": "10.0.2",
      "resolved": "https://registry.npmjs.org/remark-parse/-/remark-parse-10.0.2.tgz",
      "integrity": "sha512-3ydxgHa/ZQzG8LvC7jTXccARYDcRld3VfcgIIFs7bI6vbRSxJJmzgLEIIoYKyrfhaY+ujuWaf/PJiMZXoiCXgw==",
      "license": "MIT",
      "dependencies": {
        "@types/mdast": "^3.0.0",
        "mdast-util-from-markdown": "^1.0.0",
        "unified": "^10.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/remark-rehype": {
      "version": "10.1.0",
      "resolved": "https://registry.npmjs.org/remark-rehype/-/remark-rehype-10.1.0.tgz",
      "integrity": "sha512-EFmR5zppdBp0WQeDVZ/b66CWJipB2q2VLNFMabzDSGR66Z2fQii83G5gTBbgGEnEEA0QRussvrFHxk1HWGJskw==",
      "license": "MIT",
      "dependencies": {
        "@types/hast": "^2.0.0",
        "@types/mdast": "^3.0.0",
        "mdast-util-to-hast": "^12.1.0",
        "unified": "^10.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/rimraf": {
      "version": "5.0.10",
      "resolved": "https://registry.npmjs.org/rimraf/-/rimraf-5.0.10.tgz",
      "integrity": "sha512-l0OE8wL34P4nJH/H2ffoaniAokM2qSmrtXHmlpvYr5AVVX8msAyW0l8NVJFDxlSK4u3Uh/f41cQheDVdnYijwQ==",
      "license": "ISC",
      "dependencies": {
        "glob": "^10.3.7"
      },
      "bin": {
        "rimraf": "dist/esm/bin.mjs"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      }
    },
    "node_modules/rollup": {
      "version": "4.53.2",
      "resolved": "https://registry.npmjs.org/rollup/-/rollup-4.53.2.tgz",
      "integrity": "sha512-MHngMYwGJVi6Fmnk6ISmnk7JAHRNF0UkuucA0CUW3N3a4KnONPEZz+vUanQP/ZC/iY1Qkf3bwPWzyY84wEks1g==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "@types/estree": "1.0.8"
      },
      "bin": {
        "rollup": "dist/bin/rollup"
      },
      "engines": {
        "node": ">=18.0.0",
        "npm": ">=8.0.0"
      },
      "optionalDependencies": {
        "@rollup/rollup-android-arm-eabi": "4.53.2",
        "@rollup/rollup-android-arm64": "4.53.2",
        "@rollup/rollup-darwin-arm64": "4.53.2",
        "@rollup/rollup-darwin-x64": "4.53.2",
        "@rollup/rollup-freebsd-arm64": "4.53.2",
        "@rollup/rollup-freebsd-x64": "4.53.2",
        "@rollup/rollup-linux-arm-gnueabihf": "4.53.2",
        "@rollup/rollup-linux-arm-musleabihf": "4.53.2",
        "@rollup/rollup-linux-arm64-gnu": "4.53.2",
        "@rollup/rollup-linux-arm64-musl": "4.53.2",
        "@rollup/rollup-linux-loong64-gnu": "4.53.2",
        "@rollup/rollup-linux-ppc64-gnu": "4.53.2",
        "@rollup/rollup-linux-riscv64-gnu": "4.53.2",
        "@rollup/rollup-linux-riscv64-musl": "4.53.2",
        "@rollup/rollup-linux-s390x-gnu": "4.53.2",
        "@rollup/rollup-linux-x64-gnu": "4.53.2",
        "@rollup/rollup-linux-x64-musl": "4.53.2",
        "@rollup/rollup-openharmony-arm64": "4.53.2",
        "@rollup/rollup-win32-arm64-msvc": "4.53.2",
        "@rollup/rollup-win32-ia32-msvc": "4.53.2",
        "@rollup/rollup-win32-x64-gnu": "4.53.2",
        "@rollup/rollup-win32-x64-msvc": "4.53.2",
        "fsevents": "~2.3.2"
      }
    },
    "node_modules/sade": {
      "version": "1.8.1",
      "resolved": "https://registry.npmjs.org/sade/-/sade-1.8.1.tgz",
      "integrity": "sha512-xal3CZX1Xlo/k4ApwCFrHVACi9fBqJ7V+mwhBsuf/1IOKbBy098Fex+Wa/5QMubw09pSZ/u8EY8PWgevJsXp1A==",
      "license": "MIT",
      "dependencies": {
        "mri": "^1.1.0"
      },
      "engines": {
        "node": ">=6"
      }
    },
    "node_modules/safe-buffer": {
      "version": "5.2.1",
      "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
      "integrity": "sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ==",
      "funding": [
        {
          "type": "github",
          "url": "https://github.com/sponsors/feross"
        },
        {
          "type": "patreon",
          "url": "https://www.patreon.com/feross"
        },
        {
          "type": "consulting",
          "url": "https://feross.org/support"
        }
      ],
      "license": "MIT"
    },
    "node_modules/scheduler": {
      "version": "0.27.0",
      "resolved": "https://registry.npmjs.org/scheduler/-/scheduler-0.27.0.tgz",
      "integrity": "sha512-eNv+WrVbKu1f3vbYJT/xtiF5syA5HPIMtf9IgY/nKg0sWqzAUEvqY/xm7OcZc/qafLx/iO9FgOmeSAp4v5ti/Q==",
      "license": "MIT"
    },
    "node_modules/semver": {
      "version": "6.3.1",
      "resolved": "https://registry.npmjs.org/semver/-/semver-6.3.1.tgz",
      "integrity": "sha512-BR7VvDCVHO+q2xBEWskxS6DJE1qRnb7DxzUrogb71CWoSficBxYsiAGd+Kl0mmq/MprG9yArRkyrQxTO6XjMzA==",
      "dev": true,
      "license": "ISC",
      "bin": {
        "semver": "bin/semver.js"
      }
    },
    "node_modules/shebang-command": {
      "version": "2.0.0",
      "resolved": "https://registry.npmjs.org/shebang-command/-/shebang-command-2.0.0.tgz",
      "integrity": "sha512-kHxr2zZpYtdmrN1qDjrrX/Z1rR1kG8Dx+gkpK1G4eXmvXswmcE1hTWBWYUzlraYw1/yZp6YuDY77YtvbN0dmDA==",
      "license": "MIT",
      "dependencies": {
        "shebang-regex": "^3.0.0"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/shebang-regex": {
      "version": "3.0.0",
      "resolved": "https://registry.npmjs.org/shebang-regex/-/shebang-regex-3.0.0.tgz",
      "integrity": "sha512-7++dFhtcx3353uBaq8DDR4NuxBetBzC7ZQOhmTQInHEd6bSrXdiEyzCvG07Z44UYdLShWUyXt5M/yhz8ekcb1A==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/signal-exit": {
      "version": "4.1.0",
      "resolved": "https://registry.npmjs.org/signal-exit/-/signal-exit-4.1.0.tgz",
      "integrity": "sha512-bzyZ1e88w9O1iNJbKnOlvYTrWPDl46O1bG0D3XInv+9tkPrxrN8jUUTiFlDkkmKWgn1M6CfIA13SuGqOa9Korw==",
      "license": "ISC",
      "engines": {
        "node": ">=14"
      },
      "funding": {
        "url": "https://github.com/sponsors/isaacs"
      }
    },
    "node_modules/source-map-js": {
      "version": "1.2.1",
      "resolved": "https://registry.npmjs.org/source-map-js/-/source-map-js-1.2.1.tgz",
      "integrity": "sha512-UXWMKhLOwVKb728IUtQPXxfYU+usdybtUrK/8uGE8CQMvrhOpwvzDBwj0QhSL7MQc7vIsISBG8VQ8+IDQxpfQA==",
      "dev": true,
      "license": "BSD-3-Clause",
      "engines": {
        "node": ">=0.10.0"
      }
    },
    "node_modules/space-separated-tokens": {
      "version": "2.0.2",
      "resolved": "https://registry.npmjs.org/space-separated-tokens/-/space-separated-tokens-2.0.2.tgz",
      "integrity": "sha512-PEGlAwrG8yXGXRjW32fGbg66JAlOAwbObuqVoJpv/mRgoWDQfgH1wDPvtzWyUSNAXBGSk8h755YDbbcEy3SH2Q==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/string-width": {
      "version": "5.1.2",
      "resolved": "https://registry.npmjs.org/string-width/-/string-width-5.1.2.tgz",
      "integrity": "sha512-HnLOCR3vjcY8beoNLtcjZ5/nxn2afmME6lhrDrebokqMap+XbeW8n9TXpPDOqdGK5qcI3oT0GKTW6wC7EMiVqA==",
      "license": "MIT",
      "dependencies": {
        "eastasianwidth": "^0.2.0",
        "emoji-regex": "^9.2.2",
        "strip-ansi": "^7.0.1"
      },
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/sponsors/sindresorhus"
      }
    },
    "node_modules/string-width-cjs": {
      "name": "string-width",
      "version": "4.2.3",
      "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",
      "integrity": "sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWyWUwvSTriiZz/g==",
      "license": "MIT",
      "dependencies": {
        "emoji-regex": "^8.0.0",
        "is-fullwidth-code-point": "^3.0.0",
        "strip-ansi": "^6.0.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/string-width-cjs/node_modules/ansi-regex": {
      "version": "5.0.1",
      "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz",
      "integrity": "sha512-quJQXlTSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMlMWv37qOOb9pdJlMUEKFQ==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/string-width-cjs/node_modules/emoji-regex": {
      "version": "8.0.0",
      "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
      "integrity": "sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==",
      "license": "MIT"
    },
    "node_modules/string-width-cjs/node_modules/strip-ansi": {
      "version": "6.0.1",
      "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",
      "integrity": "sha512-Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ooI7KrPuUSztUdU5A==",
      "license": "MIT",
      "dependencies": {
        "ansi-regex": "^5.0.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/strip-ansi": {
      "version": "7.1.2",
      "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-7.1.2.tgz",
      "integrity": "sha512-gmBGslpoQJtgnMAvOVqGZpEz9dyoKTCzy2nfz/n8aIFhN/jCE/rCmcxabB6jOOHV+0WNnylOxaxBQPSvcWklhA==",
      "license": "MIT",
      "dependencies": {
        "ansi-regex": "^6.0.1"
      },
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/chalk/strip-ansi?sponsor=1"
      }
    },
    "node_modules/strip-ansi-cjs": {
      "name": "strip-ansi",
      "version": "6.0.1",
      "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",
      "integrity": "sha512-Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ooI7KrPuUSztUdU5A==",
      "license": "MIT",
      "dependencies": {
        "ansi-regex": "^5.0.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/strip-ansi-cjs/node_modules/ansi-regex": {
      "version": "5.0.1",
      "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz",
      "integrity": "sha512-quJQXlTSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMlMWv37qOOb9pdJlMUEKFQ==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/style-to-object": {
      "version": "0.3.0",
      "resolved": "https://registry.npmjs.org/style-to-object/-/style-to-object-0.3.0.tgz",
      "integrity": "sha512-CzFnRRXhzWIdItT3OmF8SQfWyahHhjq3HwcMNCNLn+N7klOOqPjMeG/4JSu77D7ypZdGvSzvkrbyeTMizz2VrA==",
      "license": "MIT",
      "dependencies": {
        "inline-style-parser": "0.1.1"
      }
    },
    "node_modules/tinyglobby": {
      "version": "0.2.15",
      "resolved": "https://registry.npmjs.org/tinyglobby/-/tinyglobby-0.2.15.tgz",
      "integrity": "sha512-j2Zq4NyQYG5XMST4cbs02Ak8iJUdxRM0XI5QyxXuZOzKOINmWurp3smXu3y5wDcJrptwpSjgXHzIQxR0omXljQ==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "fdir": "^6.5.0",
        "picomatch": "^4.0.3"
      },
      "engines": {
        "node": ">=12.0.0"
      },
      "funding": {
        "url": "https://github.com/sponsors/SuperchupuDev"
      }
    },
    "node_modules/trim-lines": {
      "version": "3.0.1",
      "resolved": "https://registry.npmjs.org/trim-lines/-/trim-lines-3.0.1.tgz",
      "integrity": "sha512-kRj8B+YHZCc9kQYdWfJB2/oUl9rA99qbowYYBtr4ui4mZyAQ2JpvVBd/6U2YloATfqBhBTSMhTpgBHtU0Mf3Rg==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/trough": {
      "version": "2.2.0",
      "resolved": "https://registry.npmjs.org/trough/-/trough-2.2.0.tgz",
      "integrity": "sha512-tmMpK00BjZiUyVyvrBK7knerNgmgvcV/KLVyuma/SC+TQN167GrMRciANTz09+k3zW8L8t60jWO1GpfkZdjTaw==",
      "license": "MIT",
      "funding": {
        "type": "github",
        "url": "https://github.com/sponsors/wooorm"
      }
    },
    "node_modules/typescript": {
      "version": "5.8.3",
      "resolved": "https://registry.npmjs.org/typescript/-/typescript-5.8.3.tgz",
      "integrity": "sha512-p1diW6TqL9L07nNxvRMM7hMMw4c5XOo/1ibL4aAIGmSAt9slTE1Xgw5KWuof2uTOvCg9BY7ZRi+GaF+7sfgPeQ==",
      "dev": true,
      "license": "Apache-2.0",
      "bin": {
        "tsc": "bin/tsc",
        "tsserver": "bin/tsserver"
      },
      "engines": {
        "node": ">=14.17"
      }
    },
    "node_modules/undici-types": {
      "version": "6.21.0",
      "resolved": "https://registry.npmjs.org/undici-types/-/undici-types-6.21.0.tgz",
      "integrity": "sha512-iwDZqg0QAGrg9Rav5H4n0M64c3mkR59cJ6wQp+7C4nI0gsmExaedaYLNO44eT4AtBBwjbTiGPMlt2Md0T9H9JQ==",
      "dev": true,
      "license": "MIT"
    },
    "node_modules/unified": {
      "version": "10.1.2",
      "resolved": "https://registry.npmjs.org/unified/-/unified-10.1.2.tgz",
      "integrity": "sha512-pUSWAi/RAnVy1Pif2kAoeWNBa3JVrx0MId2LASj8G+7AiHWoKZNTomq6LG326T68U7/e263X6fTdcXIy7XnF7Q==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0",
        "bail": "^2.0.0",
        "extend": "^3.0.0",
        "is-buffer": "^2.0.0",
        "is-plain-obj": "^4.0.0",
        "trough": "^2.0.0",
        "vfile": "^5.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/unist-util-generated": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/unist-util-generated/-/unist-util-generated-2.0.1.tgz",
      "integrity": "sha512-qF72kLmPxAw0oN2fwpWIqbXAVyEqUzDHMsbtPvOudIlUzXYFIeQIuxXQCRCFh22B7cixvU0MG7m3MW8FTq/S+A==",
      "license": "MIT",
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/unist-util-is": {
      "version": "5.2.1",
      "resolved": "https://registry.npmjs.org/unist-util-is/-/unist-util-is-5.2.1.tgz",
      "integrity": "sha512-u9njyyfEh43npf1M+yGKDGVPbY/JWEemg5nH05ncKPfi+kBbKBJoTdsogMu33uhytuLlv9y0O7GH7fEdwLdLQw==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/unist-util-position": {
      "version": "4.0.4",
      "resolved": "https://registry.npmjs.org/unist-util-position/-/unist-util-position-4.0.4.tgz",
      "integrity": "sha512-kUBE91efOWfIVBo8xzh/uZQ7p9ffYRtUbMRZBNFYwf0RK8koUMx6dGUfwylLOKmaT2cs4wSW96QoYUSXAyEtpg==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/unist-util-stringify-position": {
      "version": "3.0.3",
      "resolved": "https://registry.npmjs.org/unist-util-stringify-position/-/unist-util-stringify-position-3.0.3.tgz",
      "integrity": "sha512-k5GzIBZ/QatR8N5X2y+drfpWG8IDBzdnVj6OInRNWm1oXrzydiaAT2OQiA8DPRRZyAKb9b6I2a6PxYklZD0gKg==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/unist-util-visit": {
      "version": "4.1.2",
      "resolved": "https://registry.npmjs.org/unist-util-visit/-/unist-util-visit-4.1.2.tgz",
      "integrity": "sha512-MSd8OUGISqHdVvfY9TPhyK2VdUrPgxkUtWSuMHF6XAAFuL4LokseigBnZtPnJMu+FbynTkFNnFlyjxpVKujMRg==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0",
        "unist-util-is": "^5.0.0",
        "unist-util-visit-parents": "^5.1.1"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/unist-util-visit-parents": {
      "version": "5.1.3",
      "resolved": "https://registry.npmjs.org/unist-util-visit-parents/-/unist-util-visit-parents-5.1.3.tgz",
      "integrity": "sha512-x6+y8g7wWMyQhL1iZfhIPhDAs7Xwbn9nRosDXl7qoPTSCy0yNxnKc+hWokFifWQIDGi154rdUqKvbCa4+1kLhg==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0",
        "unist-util-is": "^5.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/update-browserslist-db": {
      "version": "1.1.4",
      "resolved": "https://registry.npmjs.org/update-browserslist-db/-/update-browserslist-db-1.1.4.tgz",
      "integrity": "sha512-q0SPT4xyU84saUX+tomz1WLkxUbuaJnR1xWt17M7fJtEJigJeWUNGUqrauFXsHnqev9y9JTRGwk13tFBuKby4A==",
      "dev": true,
      "funding": [
        {
          "type": "opencollective",
          "url": "https://opencollective.com/browserslist"
        },
        {
          "type": "tidelift",
          "url": "https://tidelift.com/funding/github/npm/browserslist"
        },
        {
          "type": "github",
          "url": "https://github.com/sponsors/ai"
        }
      ],
      "license": "MIT",
      "dependencies": {
        "escalade": "^3.2.0",
        "picocolors": "^1.1.1"
      },
      "bin": {
        "update-browserslist-db": "cli.js"
      },
      "peerDependencies": {
        "browserslist": ">= 4.21.0"
      }
    },
    "node_modules/uvu": {
      "version": "0.5.6",
      "resolved": "https://registry.npmjs.org/uvu/-/uvu-0.5.6.tgz",
      "integrity": "sha512-+g8ENReyr8YsOc6fv/NVJs2vFdHBnBNdfE49rshrTzDWOlUx4Gq7KOS2GD8eqhy2j+Ejq29+SbKH8yjkAqXqoA==",
      "license": "MIT",
      "dependencies": {
        "dequal": "^2.0.0",
        "diff": "^5.0.0",
        "kleur": "^4.0.3",
        "sade": "^1.7.3"
      },
      "bin": {
        "uvu": "bin.js"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/vfile": {
      "version": "5.3.7",
      "resolved": "https://registry.npmjs.org/vfile/-/vfile-5.3.7.tgz",
      "integrity": "sha512-r7qlzkgErKjobAmyNIkkSpizsFPYiUPuJb5pNW1RB4JcYVZhs4lIbVqk8XPk033CV/1z8ss5pkax8SuhGpcG8g==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0",
        "is-buffer": "^2.0.0",
        "unist-util-stringify-position": "^3.0.0",
        "vfile-message": "^3.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/vfile-message": {
      "version": "3.1.4",
      "resolved": "https://registry.npmjs.org/vfile-message/-/vfile-message-3.1.4.tgz",
      "integrity": "sha512-fa0Z6P8HUrQN4BZaX05SIVXic+7kE3b05PWAtPuYP9QLHsLKYR7/AlLW3NtOrpXRLeawpDLMsVkmk5DG0NXgWw==",
      "license": "MIT",
      "dependencies": {
        "@types/unist": "^2.0.0",
        "unist-util-stringify-position": "^3.0.0"
      },
      "funding": {
        "type": "opencollective",
        "url": "https://opencollective.com/unified"
      }
    },
    "node_modules/vite": {
      "version": "6.4.1",
      "resolved": "https://registry.npmjs.org/vite/-/vite-6.4.1.tgz",
      "integrity": "sha512-+Oxm7q9hDoLMyJOYfUYBuHQo+dkAloi33apOPP56pzj+vsdJDzr+j1NISE5pyaAuKL4A3UD34qd0lx5+kfKp2g==",
      "dev": true,
      "license": "MIT",
      "dependencies": {
        "esbuild": "^0.25.0",
        "fdir": "^6.4.4",
        "picomatch": "^4.0.2",
        "postcss": "^8.5.3",
        "rollup": "^4.34.9",
        "tinyglobby": "^0.2.13"
      },
      "bin": {
        "vite": "bin/vite.js"
      },
      "engines": {
        "node": "^18.0.0 || ^20.0.0 || >=22.0.0"
      },
      "funding": {
        "url": "https://github.com/vitejs/vite?sponsor=1"
      },
      "optionalDependencies": {
        "fsevents": "~2.3.3"
      },
      "peerDependencies": {
        "@types/node": "^18.0.0 || ^20.0.0 || >=22.0.0",
        "jiti": ">=1.21.0",
        "less": "*",
        "lightningcss": "^1.21.0",
        "sass": "*",
        "sass-embedded": "*",
        "stylus": "*",
        "sugarss": "*",
        "terser": "^5.16.0",
        "tsx": "^4.8.1",
        "yaml": "^2.4.2"
      },
      "peerDependenciesMeta": {
        "@types/node": {
          "optional": true
        },
        "jiti": {
          "optional": true
        },
        "less": {
          "optional": true
        },
        "lightningcss": {
          "optional": true
        },
        "sass": {
          "optional": true
        },
        "sass-embedded": {
          "optional": true
        },
        "stylus": {
          "optional": true
        },
        "sugarss": {
          "optional": true
        },
        "terser": {
          "optional": true
        },
        "tsx": {
          "optional": true
        },
        "yaml": {
          "optional": true
        }
      }
    },
    "node_modules/web-streams-polyfill": {
      "version": "3.3.3",
      "resolved": "https://registry.npmjs.org/web-streams-polyfill/-/web-streams-polyfill-3.3.3.tgz",
      "integrity": "sha512-d2JWLCivmZYTSIoge9MsgFCZrt571BikcWGYkjC1khllbTeDlGqZ2D8vD8E/lJa8WGWbb7Plm8/XJYV7IJHZZw==",
      "license": "MIT",
      "engines": {
        "node": ">= 8"
      }
    },
    "node_modules/which": {
      "version": "2.0.2",
      "resolved": "https://registry.npmjs.org/which/-/which-2.0.2.tgz",
      "integrity": "sha512-BLI3Tl1TW3Pvl70l3yq3Y64i+awpwXqsGBYWkkqMtnbXgrMD+yj7rhW0kuEDxzJaYXGjEW5ogapKNMEKNMjibA==",
      "license": "ISC",
      "dependencies": {
        "isexe": "^2.0.0"
      },
      "bin": {
        "node-which": "bin/node-which"
      },
      "engines": {
        "node": ">= 8"
      }
    },
    "node_modules/wrap-ansi": {
      "version": "8.1.0",
      "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-8.1.0.tgz",
      "integrity": "sha512-si7QWI6zUMq56bESFvagtmzMdGOtoxfR+Sez11Mobfc7tm+VkUckk9bW2UeffTGVUbOksxmSw0AA2gs8g71NCQ==",
      "license": "MIT",
      "dependencies": {
        "ansi-styles": "^6.1.0",
        "string-width": "^5.0.1",
        "strip-ansi": "^7.0.1"
      },
      "engines": {
        "node": ">=12"
      },
      "funding": {
        "url": "https://github.com/chalk/wrap-ansi?sponsor=1"
      }
    },
    "node_modules/wrap-ansi-cjs": {
      "name": "wrap-ansi",
      "version": "7.0.0",
      "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-7.0.0.tgz",
      "integrity": "sha512-YVGIj2kamLSTxw6NsZjoBxfSwsn0ycdesmc4p+Q21c5zPuZ1pl+NfxVdxPtdHvmNVOQ6XSYG4AUtyt/Fi7D16Q==",
      "license": "MIT",
      "dependencies": {
        "ansi-styles": "^4.0.0",
        "string-width": "^4.1.0",
        "strip-ansi": "^6.0.0"
      },
      "engines": {
        "node": ">=10"
      },
      "funding": {
        "url": "https://github.com/chalk/wrap-ansi?sponsor=1"
      }
    },
    "node_modules/wrap-ansi-cjs/node_modules/ansi-regex": {
      "version": "5.0.1",
      "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-5.0.1.tgz",
      "integrity": "sha512-quJQXlTSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMlMWv37qOOb9pdJlMUEKFQ==",
      "license": "MIT",
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/wrap-ansi-cjs/node_modules/ansi-styles": {
      "version": "4.3.0",
      "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",
      "integrity": "sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPlFp222wVALIheZJQSEg==",
      "license": "MIT",
      "dependencies": {
        "color-convert": "^2.0.1"
      },
      "engines": {
        "node": ">=8"
      },
      "funding": {
        "url": "https://github.com/chalk/ansi-styles?sponsor=1"
      }
    },
    "node_modules/wrap-ansi-cjs/node_modules/emoji-regex": {
      "version": "8.0.0",
      "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
      "integrity": "sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUmGE6A==",
      "license": "MIT"
    },
    "node_modules/wrap-ansi-cjs/node_modules/string-width": {
      "version": "4.2.3",
      "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",
      "integrity": "sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWyWUwvSTriiZz/g==",
      "license": "MIT",
      "dependencies": {
        "emoji-regex": "^8.0.0",
        "is-fullwidth-code-point": "^3.0.0",
        "strip-ansi": "^6.0.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/wrap-ansi-cjs/node_modules/strip-ansi": {
      "version": "6.0.1",
      "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",
      "integrity": "sha512-Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ooI7KrPuUSztUdU5A==",
      "license": "MIT",
      "dependencies": {
        "ansi-regex": "^5.0.1"
      },
      "engines": {
        "node": ">=8"
      }
    },
    "node_modules/ws": {
      "version": "8.18.3",
      "resolved": "https://registry.npmjs.org/ws/-/ws-8.18.3.tgz",
      "integrity": "sha512-PEIGCY5tSlUt50cqyMXfCzX+oOPqN0vuGqWzbcJ2xvnkzkq46oOpz7dQaTDBdfICb4N14+GARUDw2XV2N4tvzg==",
      "license": "MIT",
      "engines": {
        "node": ">=10.0.0"
      },
      "peerDependencies": {
        "bufferutil": "^4.0.1",
        "utf-8-validate": ">=5.0.2"
      },
      "peerDependenciesMeta": {
        "bufferutil": {
          "optional": true
        },
        "utf-8-validate": {
          "optional": true
        }
      }
    },
    "node_modules/yallist": {
      "version": "3.1.1",
      "resolved": "https://registry.npmjs.org/yallist/-/yallist-3.1.1.tgz",
      "integrity": "sha512-a4UGQaWPH59mOXUYnAG2ewncQS4i4F43Tv3JoAM+s2VDAmS9NsK8GpDMLrCHPksFT7h3K6TOoUNn2pb7RoXx4g==",
      "dev": true,
      "license": "ISC"
    }
  }
}
```

package.json

```
{
  "name": "echoforge",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "@google/genai": "^1.27.0",
    "prismjs": "^1.30.0",
    "react": "^19.2.0",
    "react-dom": "^19.2.0",
    "react-markdown": "^8.0.0",
    "react-simple-code-editor": "0.13.1"
  },
  "devDependencies": {
    "@types/node": "^22.14.0",
    "@types/react": "^19.2.6",
    "@vitejs/plugin-react": "^5.0.0",
    "typescript": "~5.8.2",
    "vite": "^6.2.0"
  }
}
```

tsconfig.json

```
{
  "compilerOptions": {
    "target": "ES2022",
    "experimentalDecorators": true,
    "useDefineForClassFields": false,
    "module": "ESNext",
    "lib": [
      "ES2022",
      "DOM",
      "DOM.Iterable"
    ],
    "skipLibCheck": true,
    "types": [
      "node"
    ],
    "moduleResolution": "bundler",
    "isolatedModules": true,
    "moduleDetection": "force",
    "allowJs": true,
    "jsx": "react-jsx",
    "paths": {
      "@/*": [
        "./*"
      ]
    },
    "allowImportingTsExtensions": true,
    "noEmit": true
  }
}
```

vite.config.ts

```
import path from 'path';
import { defineConfig, loadEnv } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig(({ mode }) => {
    const env = loadEnv(mode, '.', '');
    return {
      server: {
        port: 3000,
        host: '0.0.0.0',
      },
      plugins: [react()],
      define: {
        'process.env.API_KEY': JSON.stringify(env.GEMINI_API_KEY),
        'process.env.GEMINI_API_KEY': JSON.stringify(env.GEMINI_API_KEY)
      },
      resolve: {
        alias: {
          '@': path.resolve(__dirname, '.'),
        }
      },
      optimizeDeps: {
        include: ["react-markdown"]
      }
    };
});
```

AIActionMenu.tsx

```
import React, { useRef, useEffect } from 'react';
import { BookText, Bug, TestTubeDiagonal, Type, FlaskConical } from './Icon';
import { AIAction } from '../services/geminiService';

interface AIActionMenuProps {
  onAction: (action: AIAction) => void;
  onClose: () => void;
  position: { top: number; left: number };
}

const actions = [
  { id: 'explain', label: 'Explain This Code', icon: BookText },
  { id: 'add_docs', label: 'Add Docs & Types', icon: Type },
  { id: 'find_bugs', label: 'Find Potential Bugs', icon: Bug },
  { id: 'refactor', label: 'Refactor for Readability', icon: TestTubeDiagonal },
  { id: 'generate_tests', label: 'Generate Tests', icon: FlaskConical },
] as const;


const AIActionMenu: React.FC<AIActionMenuProps> = ({ onAction, onClose, position }) => {
    const menuRef = useRef<HTMLDivElement>(null);

    useEffect(() => {
        const handleClickOutside = (event: MouseEvent) => {
            if (menuRef.current && !menuRef.current.contains(event.target as Node)) {
                onClose();
            }
        };
        document.addEventListener('mousedown', handleClickOutside);
        return () => {
            document.removeEventListener('mousedown', handleClickOutside);
        };
    }, [onClose]);

    const handleActionClick = (actionId: AIAction) => {
        onAction(actionId);
        onClose();
    };

    return (
        <div
```

```
            ref={menuRef}
            className="absolute z-20 bg-surface border border-gray-700 rounded-lg shadow-2xl w-64 animate-fade-in-up"
            style={{ top: position.top, left: position.left, transform: 'translateX(-50%)' }}
        >
            <div className="p-2">
                {actions.map((action) => (
                    <button
                        key={action.id}
                        onClick={() => handleActionClick(action.id)}
                        className="w-full flex items-center text-left px-3 py-2 text-sm text-text-primary hover:bg-prim
                    >
                        <action.icon className="w-4 h-4 mr-3 text-secondary" />
                        {action.label}
                    </button>
                ))}
            </div>
            <style>{`
                @keyframes fade-in-up {
                    from { opacity: 0; transform: translate(-50%, -10px); }
                    to { opacity: 1; transform: translate(-50%, 0); }
                }
                .animate-fade-in-up {
                    animation: fade-in-up 0.15s ease-out forwards;
                }
            `}</style>
        </div>
    );
};

export default AIActionMenu;
```

AnalyzeButton.tsx

```
import React from 'react';
import { WandSparklesIcon, Spinner } from './Icon';

interface AnalyzeButtonProps {
  onClick: () => void;
  isLoading: boolean;
}

const AnalyzeButton: React.FC<AnalyzeButtonProps> = ({ onClick, isLoading }) => {
  return (
    <button
      onClick={onClick}
      disabled={isLoading}
      className="inline-flex items-center justify-center px-8 py-4 font-bold text-lg rounded-full text-white bg-gradien
    >
      {isLoading ? (
        <>
          <Spinner className="w-6 h-6 mr-3 animate-spin" />
          Analyzing...
        </>
      ) : (
        <>
          <WandSparklesIcon className="w-6 h-6 mr-3" />
          Analyze & Refactor
        </>
      )}
    </button>
  );
};

export default AnalyzeButton;
```

CodeDisplay.tsx

```
import React, { useState, useRef, useEffect, useMemo } from 'react';
import Editor from 'react-simple-code-editor';
import Prism from 'prismjs';
import 'prismjs/components/prism-python';
import 'prismjs/themes/prism-tomorrow.css';
import { WandSparklesIcon, PackageIcon, PlayIcon, Spinner, BugPlayIcon } from './Icon';
import AIActionMenu from './AIActionMenu';
import { AIAction } from '../services/geminiService';

interface LintError {
  line: number;
  column: number;
  end_line: number;
  end_column: number;
  message: string;
  code: string;
}

interface CodeDisplayProps {
  code: string;
  onCodeChange: (newCode: string) => void;
  onContextualAnalyze: (selectedCode: string, action: AIAction) => void;
  onAnalyzeDependencies: () => void;
  onExecuteCode: () => void;
  isExecuting: boolean;
  breakpoints: Set<number>;
  onToggleBreakpoint: (lineNumber: number) => void;
}

const useDebounce = (value: string, delay: number) => {
    const [debouncedValue, setDebouncedValue] = useState(value);
    useEffect(() => {
        const handler = setTimeout(() => {
            setDebouncedValue(value);
        }, delay);
        return () => {
            clearTimeout(handler);
        };
    }, [value, delay]);
    return debouncedValue;
};


const CodeDisplay: React.FC<CodeDisplayProps> = (props) => {
  const {
    code, onCodeChange, onContextualAnalyze, onAnalyzeDependencies,
    onExecuteCode, isExecuting, breakpoints, onToggleBreakpoint
  } = props;

  const [selection, setSelection] = useState<string | null>(null);
  const [buttonPosition, setButtonPosition] = useState<{ top: number, left: number } | null>(null);
  const [isMenuOpen, setIsMenuOpen] = useState(false);
  const editorRef = useRef<any>(null);
  const gutterRef = useRef<HTMLDivElement>(null);

  const [lintErrors, setLintErrors] = useState<LintError[]>([]);
  const linterWorkerRef = useRef<Worker | null>(null);
  const debouncedCode = useDebounce(code, 500);

  useEffect(() => {
    linterWorkerRef.current = new Worker('/linter.js');
    linterWorkerRef.current.onmessage = (event) => {
        if (event.data.type === 'lintResults') {
            setLintErrors(event.data.errors);
        }
        if (event.data.type === 'ready') {
            // Initial lint on load
            linterWorkerRef.current?.postMessage({ code });
        }
    };
    return () => {
        linterWorkerRef.current?.terminate();
    };
  }, []);

  useEffect(() => {
    if (debouncedCode && linterWorkerRef.current) {
        linterWorkerRef.current.postMessage({ code: debouncedCode });
    }
  }, [debouncedCode]);

  const highlight = (code: string) => {
    const highlighted = Prism.highlight(code, Prism.languages.python, 'python');
    const lines = highlighted.split('\n');

    lintErrors.forEach(error => {
        if (error.line <= lines.length) {
            const lineIndex = error.line - 1;
            let lineContent = lines[lineIndex];

            const squiggle = `<span class="squiggle" title="${error.message} (${error.code})">`;
            const endSquiggle = `</span>`;

            if (!lineContent.includes('<span class="squiggle"')) {
                 lines[lineIndex] = `${squiggle}${lineContent}${endSquiggle}`;
            }
        }
    });

    return lines.join('\n');
  };

  const handleSelection = () => {
    if (isMenuOpen) return;

    const currentSelection = window.getSelection()?.toString();
    if (currentSelection && currentSelection.trim().length > 0) {
      setSelection(currentSelection);
      const range = window.getSelection()?.getRangeAt(0);
      if (range) {
        const rect = range.getBoundingClientRect();
        const editorContainerRect = editorRef.current?._input.parentElement.getBoundingClientRect();
        if(editorContainerRect) {
            setButtonPosition({
              top: rect.bottom - editorContainerRect.top + 5,
              left: rect.left - editorContainerRect.left + rect.width / 2,
            });
        }
      }
    } else {
      setSelection(null);
      setButtonPosition(null);
    }
  };

  useEffect(() => {
    const editorElement = editorRef.current?._input;
    if (editorElement) {
        const handleInteraction = () => handleSelection();
        editorElement.addEventListener('mouseup', handleInteraction);
        editorElement.addEventListener('keyup', handleInteraction);

        const syncScroll = () => {
            if (gutterRef.current) {
                gutterRef.current.scrollTop = editorElement.scrollTop;
            }
        };
        editorElement.addEventListener('scroll', syncScroll);

        return () => {
            editorElement.removeEventListener('mouseup', handleInteraction);
            editorElement.removeEventListener('keyup', handleInteraction);
            editorElement.removeEventListener('scroll', syncScroll);
        };
    }
  }, [editorRef.current, isMenuOpen]);

  const handleWandClick = () => {
    setIsMenuOpen(true);
  };

  const handleMenuClose = () => {
    setIsMenuOpen(false);
    setSelection(null);
    setButtonPosition(null);
  };

  const handleActionSelect = (action: AIAction) => {
    if (selection) {
        onContextualAnalyze(selection, action);
    }
  };

  const lineCount = useMemo(() => code.split('\n').length, [code]);
  const isDebugging = breakpoints.size > 0;

  return (
    <div className="bg-surface rounded-lg shadow-lg flex flex-col overflow-hidden h-full">
      <div className="p-3 bg-gray-900/50 border-b border-gray-700 flex justify-between items-center">
        <h2 className="text-lg font-semibold text-text-secondary tracking-wide">Python IDE</h2>
        <div className="flex items-center space-x-2">
            <button
                onClick={onAnalyzeDependencies}
                className="flex items-center space-x-2 px-3 py-1 text-sm bg-primary/40 text-text-primary rounded-md hov
                title="Scan for third-party dependencies"
            >
                <PackageIcon className="w-4 h-4" />
                <span>Scan Dependencies</span>
            </button>
            <button
                onClick={onExecuteCode}
                disabled={isExecuting}
                className={`flex items-center space-x-2 px-3 py-1 text-sm text-white rounded-md transition-colors disab
                title={isDebugging ? "Run script with debugger" : "Run the script"}
            >
                {isExecuting ? <Spinner className="w-4 h-4 animate-spin" /> : (isDebugging ? <BugPlayIcon className="w-
                <span>{isExecuting ? 'Running...' : (isDebugging ? 'Debug Script' : 'Run Script')}</span>
            </button>
        </div>
      </div>
      <div className="overflow-hidden flex-grow relative font-mono text-sm editor-container flex">
        <div ref={gutterRef} className="gutter bg-gray-900/50 text-right pr-2 select-none overflow-hidden" style={{font
           {Array.from({ length: lineCount }, (_, i) => i + 1).map(lineNumber => (
               <div key={lineNumber} className="relative text-gray-500 h-[21px] flex items-center justify-end" onClick=
                    {breakpoints.has(lineNumber) && <div className="absolute left-1 w-2 h-2 bg-red-500 rounded-full"></
                    {lineNumber}
               </div>
           ))}
        </div>
        <div className="flex-grow relative">
            {buttonPosition && !isMenuOpen && (
              <button
                className="absolute z-10 bg-secondary text-white rounded-full p-2 shadow-lg hover:bg-green-500 transiti
                style={{ top: buttonPosition.top, left: buttonPosition.left, transform: 'translateX(-50%)' }}
                aria-label="AI Actions"
                onClick={handleWandClick}
              >
                <WandSparklesIcon className="w-5 h-5" />
              </button>
            )}
            {isMenuOpen && buttonPosition && (
                <AIActionMenu
```

```
                    position={{ top: buttonPosition.top + 10, left: buttonPosition.left }}
                    onAction={handleActionSelect}
                    onClose={handleMenuClose}
                />
            )}
            <Editor
              ref={editorRef}
              value={code}
              onValueChange={onCodeChange}
              highlight={highlight}
              padding={16}
              className="bg-surface text-gray-300 w-full h-full focus:outline-none resize-none whitespace-pre break-wor
              style={{
                fontFamily: '"Fira Code", monospace',
                fontSize: 14,
                lineHeight: '21px',
                outline: 'none',
                border: 'none',
              }}
              aria-label="Python code editor"
            />
        </div>
      </div>
       <style>{`
        .editor-container > .gutter {
            flex-shrink: 0;
            padding-top: 16px;
            padding-bottom: 16px;
        }
        .editor-container > .gutter div {
            cursor: pointer;
        }
        .editor-container > textarea:focus-visible, .editor-container > pre:focus-visible {
            outline: none;
        }
        .token.comment,
        .token.prolog,
        .token.doctype,
        .token.cdata {
            color: #999;
        }
        .squiggle {
          text-decoration: underline;
          text-decoration-style: wavy;
          text-decoration-color: #ef4444; /* red-500 */
          text-decoration-skip-ink: none;
          position: relative;
          cursor: help;
        }
       `}</style>
    </div>
  );
};

export default CodeDisplay;
```

DependencyViewer.tsx

```
import React, { useState } from 'react';
import { Dependency } from '../services/geminiService';
import { Spinner, PackageIcon } from './Icon';

interface DependencyViewerProps {
    dependencies: Dependency[] | null;
    isLoading: boolean;
    error: string | null;
    onGenerate: () => void;
    requirementsFile: string | null;
}
```

```
const DependencyViewer: React.FC<DependencyViewerProps> = ({ dependencies, isLoading, error, onGenerate, requirementsFi
    const [isCopied, setIsCopied] = useState(false);

    const handleCopy = () => {
        if (!requirementsFile) return;
        navigator.clipboard.writeText(requirementsFile).then(() => {
            setIsCopied(true);
            setTimeout(() => setIsCopied(false), 2000);
        });
    };

    if (isLoading) {
        return (
          <div className="flex flex-col items-center justify-center h-full text-text-secondary">
            <Spinner className="h-12 w-12 animate-spin mb-4 text-secondary" />
            <p className="text-lg font-medium">Analyzing dependencies...</p>
            <p className="text-sm text-gray-500">The AI is scanning your imports.</p>
          </div>
        );
    }

    if (error) {
        return (
          <div className="flex items-center justify-center h-full text-red-400">
            <div className="text-center p-4 bg-red-900/20 rounded-lg">
              <p className="font-bold text-lg">An Error Occurred</p>
              <p>{error}</p>
            </div>
          </div>
        );
    }

    if (!dependencies) {
        return (
          <div className="flex flex-col items-center justify-center h-full text-text-secondary p-8 text-center">
              <PackageIcon className="w-16 h-16 mb-4 text-gray-600"/>
              <h3 className="text-2xl font-bold mb-2">Project Dependencies</h3>
              <p className="max-w-md">Click "Scan Dependencies" in the IDE header to analyze your code and list all thi
          </div>
        );
    }

    if (dependencies.length === 0) {
        return (
             <div className="flex flex-col items-center justify-center h-full text-text-secondary p-8 text-center">
              <PackageIcon className="w-16 h-16 mb-4 text-gray-600"/>
              <h3 className="text-2xl font-bold mb-2">No Third-Party Dependencies Found</h3>
              <p className="max-w-md">The AI analysis did not find any external libraries in your code. Only Python sta
          </div>
        )
    }

    return (
        <div className="h-full flex flex-col">
            <div className="flex-grow overflow-y-auto">
                <div className="flex justify-between items-center mb-4">
                    <h3 className="text-xl font-semibold">Detected Dependencies ({dependencies.length})</h3>
                    <button
                        onClick={onGenerate}
                        className="px-4 py-2 text-sm font-medium text-white bg-secondary rounded-md hover:opacity-80 tr
                        disabled={!!requirementsFile}
                    >
                        {requirementsFile ? 'Generated!' : 'Generate requirements.txt'}
                    </button>
                </div>

                <div className="space-y-3">
                    {dependencies.map((dep, index) => (
                        <div key={index} className="p-4 bg-background/50 rounded-lg border border-gray-700">
                            <div className="flex items-center space-x-3">
```

```
                                <PackageIcon className="w-6 h-6 text-secondary" />
                                <div>
                                    <p className="font-mono font-bold text-text-primary">{dep.packageName}</p>
                                    <p className="text-xs text-text-secondary">from import: <span className="font-mono"
                                </div>
                            </div>
                            <p className="mt-2 text-sm text-text-secondary italic">"{dep.justification}"</p>
                        </div>
                    ))}
                </div>
            </div>

            {requirementsFile && (
                <div className="mt-6 pt-4 border-t border-gray-700">
                    <h4 className="font-semibold mb-2">Generated requirements.txt</h4>
                    <div className="relative bg-background rounded-md">
                        <pre className="p-4 text-sm font-mono whitespace-pre-wrap overflow-x-auto">
                            {requirementsFile}
                        </pre>
                        <button
                            onClick={handleCopy}
                            className="absolute top-2 right-2 px-3 py-1 text-xs bg-gray-600 hover:bg-gray-500 rounded-m
                        >
                            {isCopied ? 'Copied!' : 'Copy'}
                        </button>
                    </div>
                </div>
            )}
        </div>
    );
};

export default DependencyViewer;
```

DiffViewer.tsx

```
import React from 'react';
import Prism from 'prismjs';
import 'prismjs/components/prism-python';

interface DiffViewerProps {
  oldCode: string;
  newCode: string;
}

// A simple line-based diff algorithm
const createDiff = (oldStr: string, newStr: string) => {
    const oldLines = oldStr.split('\n');
    const newLines = newStr.split('\n');
    const oldSet = new Set(oldLines);
    const newSet = new Set(newLines);

    const removed = oldLines.filter(line => !newSet.has(line)).map(line => ({ type: 'removed' as const, content: line }
    const added = newLines.filter(line => !oldSet.has(line)).map(line => ({ type: 'added' as const, content: line }));
    const common = oldLines.filter(line => newSet.has(line)).map(line => ({ type: 'common' as const, content: line }));

    // This is a simplified view. A more complex diff would interleave lines.
    // For this UI, showing removed then common then added is clear enough.
    return [...removed, ...common, ...added];
};

const DiffViewer: React.FC<DiffViewerProps> = ({ oldCode, newCode }) => {
    const diff = createDiff(oldCode, newCode);

    return (
        <pre className="text-sm font-mono bg-gray-900/70 text-gray-300 w-full focus:outline-none resize-none whitespace
            {diff.map((line, index) => {
                let lineClass = '';
                let symbol = '';
                switch (line.type) {
                    case 'added':
                        lineClass = 'bg-green-900/40';
                        symbol = '+ ';
                        break;
                    case 'removed':
                        lineClass = 'bg-red-900/40';
                        symbol = '- ';
                        break;
                    default:
                        symbol = '  ';
                        break;
                }
                const highlightedCode = Prism.highlight(line.content, Prism.languages.python, 'python');
                return (
                    <div key={index} className={`flex ${lineClass}`}>
                        <span className="w-6 text-center select-none text-gray-500">{symbol}</span>
                        <code className="flex-1" dangerouslySetInnerHTML={{ __html: highlightedCode }} />
                    </div>
                )
            })}
        </pre>
    );
};

export default DiffViewer;
```

ExecutionConsole.tsx

```
import React, { useState, useEffect, useRef, useMemo } from 'react';
import { Spinner, Trash2Icon, ChevronUpIcon, ChevronDownIcon, LightbulbIcon } from './Icon';

interface ExecutionConsoleProps {
  output: { type: 'stdout' | 'stderr' | 'error', message: string }[];
  isExecuting: boolean;
  onClear: () => void;
  onExplainError: (traceback: string) => void;
}

const ExecutionConsole: React.FC<ExecutionConsoleProps> = ({ output, isExecuting, onClear, onExplainError }) => {
  const [isCollapsed, setIsCollapsed] = useState(true);
  const outputEndRef = useRef<HTMLDivElement>(null);

  useEffect(() => {
    outputEndRef.current?.scrollIntoView({ behavior: 'smooth' });
  }, [output]);

  useEffect(() => {
    if (isExecuting) {
      setIsCollapsed(false);
    }
  }, [isExecuting]);

  const traceback = useMemo(() => {
    const errorOutput = output.filter(line => line.type === 'stderr' || line.type === 'error').map(line => line.message
    return errorOutput.includes('Traceback (most recent call last):') ? errorOutput : null;
  }, [output]);

  return (
    <div className={`bg-surface rounded-lg shadow-lg flex flex-col transition-all duration-300 ${isCollapsed ? 'h-12' :
        <div
            className="p-2 bg-gray-900/50 border-b border-gray-700 flex justify-between items-center cursor-pointer"
            onClick={() => setIsCollapsed(!isCollapsed)}
        >
            <h3 className="text-md font-semibold text-text-secondary tracking-wide">
                Output Console
            </h3>
            <div className="flex items-center space-x-3">
                {traceback && !isExecuting && (
                     <button
                        onClick={(e) => { e.stopPropagation(); onExplainError(traceback); }}
                        className="flex items-center space-x-1.5 px-2 py-0.5 text-xs bg-amber-600/80 text-white rounded
                        title="Let AI explain this error"
                    >
                        <LightbulbIcon className="w-3.5 h-3.5" />
                        <span>Explain Error</span>
                    </button>
                )}
                {isExecuting && <Spinner className="w-4 h-4 animate-spin text-secondary" />}
                <button
                    onClick={(e) => { e.stopPropagation(); onClear(); }}
                    className="text-text-secondary hover:text-text-primary disabled:opacity-50"
                    title="Clear console"
                    disabled={output.length === 0}
                >
                    <Trash2Icon className="w-4 h-4" />
                </button>
                <button className="text-text-secondary hover:text-text-primary">
                    {isCollapsed ? <ChevronUpIcon className="w-5 h-5" /> : <ChevronDownIcon className="w-5 h-5" />}
                </button>
            </div>
        </div>
        {!isCollapsed && (
            <div className="p-4 overflow-auto flex-grow font-mono text-sm text-gray-300">
                {output.length === 0 && !isExecuting && (
                    <p className="text-gray-500">Click "Run Script" to see the output here.</p>
                )}
                {output.map((line, index) => (
                    <div key={index} className={`whitespace-pre-wrap ${line.type === 'stderr' || line.type === 'error'
                        <span className="select-none mr-2 text-gray-600">&gt;</span>
                        {line.message}
                    </div>
                ))}
                 {isExecuting && output.length === 0 && (
                    <p className="text-gray-400">Executing code...</p>
                 )}
                <div ref={outputEndRef} />
            </div>
        )}
    </div>
  );
};

export default ExecutionConsole;
```

ExtensionGuideModal.tsx

```
import React from 'react';
import { GraduationCapIcon } from './Icon';

interface ExtensionGuideModalProps {
  isOpen: boolean;
  onClose: () => void;
}

const ExtensionGuideModal: React.FC<ExtensionGuideModalProps> = ({ isOpen, onClose }) => {
  if (!isOpen) return null;

  return (
    <div
      className="fixed inset-0 bg-black/60 z-50 flex justify-center items-center"
      onClick={onClose}
    >
      <div
        className="bg-surface rounded-lg shadow-2xl w-full max-w-4xl h-[80vh] border border-gray-700 animate-fade-in fl
        onClick={(e) => e.stopPropagation()}
      >
        <div className="p-6 border-b border-gray-700 flex items-center justify-between">
          <div className="flex items-center space-x-3">
            <GraduationCapIcon className="h-7 w-7 text-secondary" />
            <h2 className="text-xl font-bold">Become a Dev: AI-Guided Extension Builder</h2>
          </div>
          <button onClick={onClose} className="text-text-secondary hover:text-text-primary">&times;</button>
        </div>
        <div className="p-8 flex-grow overflow-y-auto">
          <div className="prose prose-invert max-w-none">
            <h1 className="text-secondary">Welcome, Pioneer!</h1>
            <p>
              This is your first step towards extending the capabilities of this AI Code Refactor IDE.
              We believe that the best tools are the ones you can shape yourself. This guide will walk you through
              the process of creating your very own extensions, powered by AI.
            </p>
            <p>
              You'll learn how to add new actions to the "Magic Wand" menu, create custom AI prompts, and integrate
              your own ideas directly into the application's workflow.
            </p>
            <div className="bg-primary/20 border border-primary/50 p-4 rounded-lg my-6">
                <h3 className="font-bold text-lg">Coming Soon: Interactive AI Tutorial</h3>
                <p>
                    This space will soon feature a step-by-step, interactive tutorial. You'll be able to write code,
                    get instant feedback from an AI tutor, and see your extension come to life in real-time.
                </p>
                <p>Stay tuned for this exciting feature!</p>
            </div>
          </div>
        </div>
        <div className="px-6 py-4 bg-background/50 flex justify-end space-x-3 rounded-b-lg border-t border-gray-700">
          <button
            onClick={onClose}
            className="px-5 py-2 text-sm font-medium text-white bg-secondary rounded-md hover:opacity-80 transition"
          >
            Close
          </button>
        </div>
      </div>
       <style>{`
        @keyframes fade-in {
            from { opacity: 0; transform: scale(0.95); }
            to { opacity: 1; transform: scale(1); }
        }
        .animate-fade-in {
            animation: fade-in 0.2s ease-out forwards;
        }
      `}</style>
    </div>
  );
};

export default ExtensionGuideModal;
```

FileGraph.css

```
/* Basic styles for FileGraph */

.file-graph-container {
    padding: 1rem;
    border: 1px solid #ccc;
    border-radius: 8px;
    background-color: #f9f9f9;
}

.file-graph-title {
    font-size: 1.5rem;
    margin-bottom: 1rem;
}

.file-graph-list {
    list-style: none;
    padding: 0;
}
```

```
.file-graph-node {
    padding: 0.5rem;
    border: 1px solid #ddd;
    border-radius: 4px;
    margin-bottom: 0.5rem;
    cursor: pointer;
    display: flex;
    align-items: center;
    gap: 0.5rem;
}

.file-graph-node:hover {
    background-color: #eef;
}
```

FileGraph.tsx

```
import React from "react";
import fileLinks from "../file_links.json";
import "./FileGraph.css";

const getIcon = (type: string) => {
    if (type === "folder") return "■";
    if (type === "file") return "■";
    return "■";
};

export const FileGraph: React.FC = () => {
    const handleClick = (file: { name: string; type: string }) => {
        alert(`Name: ${file.name}\nType: ${file.type}`);
    };
    return (
        <div className="file-graph-container">
            <h2 className="file-graph-title">File Graph (Prototype)</h2>
            <div className="file-graph-list">
                {fileLinks.files.map((file: { name: string; type: string }, i: number) => (
                    <div
                        key={i}
                        className="file-graph-node"
                        onClick={() => handleClick(file)}
                        style={{ cursor: "pointer", display: "flex", alignItems: "center", gap: "0.5em" }}
                    >
                        <span>{getIcon(file.type)}</span>
                        <span>{file.name}</span>
                    </div>
                ))}
            </div>
        </div>
    );
}
```

GeneratedTestViewer.tsx

```
import React, { useState } from 'react';
import ReactMarkdown from 'https://esm.sh/react-markdown@9';
import remarkGfm from 'https://esm.sh/remark-gfm@4';
import Prism from 'prismjs';
import 'prismjs/components/prism-python';

interface GeneratedTestViewerProps {
    explanation: string;
    code: string;
}

const GeneratedTestViewer: React.FC<GeneratedTestViewerProps> = ({ explanation, code }) => {
    const [isCopied, setIsCopied] = useState(false);

    const handleCopy = () => {
        navigator.clipboard.writeText(code).then(() => {
            setIsCopied(true);
            setTimeout(() => setIsCopied(false), 2000);
        });
    };

    const highlightedCode = Prism.highlight(code, Prism.languages.python, 'python');

    return (
        <div>
            <div className="prose prose-invert max-w-none mb-6">
                <ReactMarkdown children={explanation} remarkPlugins={[remarkGfm]} />
            </div>
            <h3 className="text-lg font-semibold text-text-secondary mb-2">Generated Test Code:</h3>
            <div className="relative bg-gray-900/70 rounded-lg border border-surface">
                <pre className="p-4 text-sm font-mono overflow-x-auto">
                    <code dangerouslySetInnerHTML={{ __html: highlightedCode }} />
                </pre>
                <button
                    onClick={handleCopy}
                    className="absolute top-2 right-2 px-3 py-1 text-xs bg-gray-600 hover:bg-gray-500 rounded-md text-w
                >
                    {isCopied ? 'Copied!' : 'Copy Code'}
                </button>
            </div>
        </div>
    );
};

export default GeneratedTestViewer;
```

Header.tsx

```
import React, { useState } from 'react';
import { BrainCircuitIcon, SettingsIcon, GraduationCapIcon } from './Icon';
import SettingsModal from './SettingsModal';
import ExtensionGuideModal from './ExtensionGuideModal';

const Header: React.FC = () => {
  const [isSettingsOpen, setIsSettingsOpen] = useState(false);
  const [isGuideOpen, setIsGuideOpen] = useState(false);

  return (
    <>
      <header className="bg-surface p-4 border-b border-gray-700 shadow-md">
        <div className="max-w-7xl mx-auto flex items-center justify-between">
            <div className="flex items-center space-x-3">
                <BrainCircuitIcon className="h-8 w-8 text-secondary" />
                <h1 className="text-2xl font-bold text-text-primary tracking-tight">
                AI Code Refactor
                </h1>
                <span className="text-xs font-mono bg-primary text-blue-200 px-2 py-1 rounded">for Python</span>
            </div>
            <div className="flex items-center space-x-2">
                <button
                    onClick={() => setIsGuideOpen(true)}
                    className="p-2 rounded-full text-text-secondary hover:bg-gray-700 hover:text-text-primary transitio
                    aria-label="Open extension guide"
                >
                    <GraduationCapIcon className="h-6 w-6" />
                </button>
                <button
                    onClick={() => setIsSettingsOpen(true)}
                    className="p-2 rounded-full text-text-secondary hover:bg-gray-700 hover:text-text-primary transitio
                    aria-label="Open settings"
                >
                    <SettingsIcon className="h-6 w-6" />
                </button>
            </div>
        </div>
      </header>
      <SettingsModal
```

```
        isOpen={isSettingsOpen}
        onClose={() => setIsSettingsOpen(false)}
      />
      <ExtensionGuideModal
        isOpen={isGuideOpen}
        onClose={() => setIsGuideOpen(false)}
      />
    </>
  );
};

export default Header;
```

Icon.tsx

```
import React from 'react';

interface IconProps {
    className?: string;
}

export const BrainCircuitIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" className={className} width="24" height="24" viewBox="0 0 24 24" fill="none
        <path d="M12 5a3 3 0 1 0-5.993.142"/>
        <path d="M18 5a3 3 0 1 0-5.993.142"/>
        <path d="M12 19a3 3 0 1 0-5.993-.142"/>
        <path d="M18 19a3 3 0 1 0-5.993-.142"/>
        <path d="M12 12a3 3 0 1 0-5.993.142"/>
        <path d="M18 12a3 3 0 1 0-5.993.142"/>
        <path d="M6 7.857v8.286"/>
        <path d="M18 7.857v8.286"/>
        <path d="M9.003 5.142A2.993 2.993 0 0 1 12 5"/>
        <path d="M15 5a2.993 2.993 0 0 1 2.997.142"/>
        <path d="M9.003 18.858A2.993 2.993 0 0 0 12 19"/>
        <path d="M15 19a2.993 2.993 0 0 0 2.997-.142"/>
        <path d="M9.003 12a2.993 2.993 0 0 1 2.997-.142"/>
        <path d="M15 12a2.993 2.993 0 0 1 2.997-.142"/>
    </svg>
);

export const WandSparklesIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" className={className} width="24" height="24" viewBox="0 0 24 24" fill="none
        <path d="m5 3 2.5 5L10 3"/>
        <path d="m14 3 2.5 5L19 3"/>
        <path d="M10 22v-5"/>
        <path d="M14 22v-5"/>
        <path d="M17 10h.01"/>
        <path d="M7 10h.01"/>
        <path d="m7 16 2.5-5L12 16"/>
        <path d="m12 16 2.5-5L17 16"/>
    </svg>
);

export const Spinner: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" className={className} width="24" height="24" viewBox="0 0 24 24" fill="none
        <path d="M21 12a9 9 0 1 1-6.219-8.56" />
    </svg>
);

export const BookText: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
      <path d="M4 19.5v-15A2.5 2.5 0 0 1 6.5 2H20v20H6.5a2.5 2.5 0 0 1 0-5H20"/>
      <path d="M8 7h6"/>
      <path d="M8 11h8"/>
    </svg>
);

export const Bug: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
      <path d="M20 9V7a2 2 0 0 0-2-2h-4.5a2 2 0 0 0-1.9.9L9 9"/>
      <path d="M12 18v-4"/>
      <path d="M6 10H4a2 2 0 0 0-2 2v4a2 2 0 0 0 2 2h4.5a2 2 0 0 0 1.9-.9L15 14"/>
      <path d="M20 16h-4a2 2 0 0 0-2 2v0a2 2 0 0 0 2 2h4v-4Z"/>
      <path d="M4 12h4"/>
      <path d="M9 18l-3.5-3.5"/>
      <path d="m15 10 3.5 3.5"/>
    </svg>
);

export const Type: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
      <path d="M4 7V4h16v3"/>
      <path d="M9 20h6"/>
      <path d="M12 4v16"/>
    </svg>
);

export const TestTubeDiagonal: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
      <path d="M14.5 2.5 21.5 9.5"/>
      <path d="M6.31 20.18c-1.63-1.62-2.43-4.2-1.2-6.5l5.4-5.4c2.3-2.3 5.3-3 7.5-1.7l-2.1 2.1c-.5-.4-1.2-.5-1.8-.2l-2.6
    </svg>
);

export const SettingsIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M12.22 2h-.44a2 2 0 0 0-2 2v.18a2 2 0 0 1-1 1.73l-.43.25a2 2 0 0 1-2 0l-.15-.08a2 2 0 0 0-2.73.73l-.22
        <circle cx="12" cy="12" r="3"/>
    </svg>
);

export const GraduationCapIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M21.42 10.922a1 1 0 0 0-.019-1.838L12.83 5.18a2 2 0 0 0-1.66 0L2.6 9.08a1 1 0 0 0 0 1.832l8.57 3.908a2
        <path d="M22 10v6"/>
        <path d="M6 12.5V16a6 3 0 0 0 12 0v-3.5"/>
    </svg>
);

export const PackageIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M16.5 9.4a4.5 4.5 0 1 1-9 0a4.5 4.5 0 0 1 9 0z"/>
        <path d="M12 14.1c-2.7 0-4.5-1.3-4.5-3.2s1.8-3.2 4.5-3.2 4.5 1.3 4.5 3.2-1.8 3.2-4.5 3.2z"/>
        <path d="M21 16V8a2 2 0 0 0-1-1.73l-7-4a2 2 0 0 0-2 0l-7 4A2 2 0 0 0 3 8v8a2 2 0 0 0 1 1.73l7 4a2 2 0 0 0 2 0l7
        <path d="m3.3 7 8.7 5 8.7-5"/>
        <path d="M12 22V12"/>
    </svg>
);

export const FlaskConical: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M10 2v7.31"/>
        <path d="M14 9.31V2"/>
        <path d="M3 13h18"/>
        <path d="M4 13a8 8 0 0 0 16 0"/>
    </svg>
);

export const PlayIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <polygon points="5 3 19 12 5 21 5 3"/>
    </svg>
);

export const Trash2Icon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M3 6h18"/>
        <path d="M19 6v14a2 2 0 0 1-2 2H7a2 2 0 0 1-2-2V6m3 0V4a2 2 0 0 1 2-2h4a2 2 0 0 1 2 2v2"/>
        <path d="M10 11v6"/>
        <path d="M14 11v6"/>
    </svg>
);

export const ChevronUpIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="m18 15-6-6-6 6"/>
    </svg>
);

export const ChevronDownIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="m6 9 6 6 6-6"/>
    </svg>
);

export const LightbulbIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
      <path d="M15 14c.2-1 .7-1.7 1.5-2.5 1-.9 1.5-2.2 1.5-3.5A6 6 0 0 0 6 8c0 1 .2 2.2 1.5 3.5.7.7 1.3 1.5 1.5 2.5"/>
      <path d="M9 18h6"/>
      <path d="M10 22h4"/>
    </svg>
);

export const HistoryIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M3 12a9 9 0 1 0 9-9 9.75 9.75 0 0 0-6.74 2.74L3 8"/>
        <path d="M3 3v5h5"/>
        <path d="M12 7v5l4 2"/>
    </svg>
);

export const BugPlayIcon: React.FC<IconProps> = ({ className }) => (
    <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor
        <path d="M12 12h.01"/>
        <path d="M20 9V7a2 2 0 0 0-2-2h-4.5a2 2 0 0 0-1.9.9L9 9"/>
        <path d="M12 18v-4"/>
        <path d="M6 10H4a2 2 0 0 0-2 2v4a2 2 0 0 0 2 2h4.5a2 2 0 0 0 1.9-.9L15 14"/>
        <path d="M20 16h-4a2 2 0 0 0-2 2v0a2 2 0 0 0 2 2h4v-4Z"/>
        <path d="M4 12h4"/>
        <path d="M9 18l-3.5-3.5"/>
        <path d="m15 10 3.5 3.5"/>
    </svg>
);
```

MarkdownViewer.css

```
/* Styles for MarkdownViewer */

.markdown-viewer {
    display: flex;
    gap: 1rem;
    padding: 1rem;
    border: 1px solid #ccc;
    border-radius: 8px;
    background-color: #f9f9f9;
}

.markdown-editor {
    width: 50%;
    height: 300px;
    padding: 0.5rem;
    font-family: monospace;
    font-size: 1rem;
    border: 1px solid #ddd;
    border-radius: 4px;
    resize: vertical;
}

.markdown-preview {
    width: 50%;
    height: 300px;
    padding: 0.5rem;
    border: 1px solid #ddd;
    border-radius: 4px;
    background-color: #fff;
    overflow-y: auto;
}
```

MarkdownViewer.tsx

```
import React, { useState } from "react";
import ReactMarkdown from "react-markdown";
import "./MarkdownViewer.css";

interface MarkdownViewerProps {
    fileContent: string;
}

export const MarkdownViewer: React.FC<MarkdownViewerProps> = ({ fileContent }) => {
    const [content, setContent] = useState(fileContent);

    const handleEdit = (e: React.ChangeEvent<HTMLTextAreaElement>) => {
        setContent(e.target.value);
    };

    return (
        <div className="markdown-viewer">
            <textarea
                className="markdown-editor"
                value={content}
                onChange={handleEdit}
                placeholder="Write your markdown here..."
            />
            <div className="markdown-preview">
                <ReactMarkdown>{content}</ReactMarkdown>
            </div>
        </div>
    );
};
```

RevisionHistoryViewer.tsx

```
import React, { useState } from 'react';
import { Revision } from '../App';
import { HistoryIcon } from './Icon';
import DiffViewer from './DiffViewer';

interface RevisionHistoryViewerProps {
    revisions: Revision[];
    currentCode: string;
    onRestore: (code: string) => void;
}

const timeAgo = (timestamp: number): string => {
    const now = new Date();
    const seconds = Math.floor((now.getTime() - timestamp) / 1000);
    let interval = seconds / 31536000;
    if (interval > 1) return Math.floor(interval) + " years ago";
    interval = seconds / 2592000;
    if (interval > 1) return Math.floor(interval) + " months ago";
    interval = seconds / 86400;
    if (interval > 1) return Math.floor(interval) + " days ago";
    interval = seconds / 3600;
    if (interval > 1) return Math.floor(interval) + " hours ago";
    interval = seconds / 60;
    if (interval > 1) return Math.floor(interval) + " minutes ago";
    return "just now";
}
```

```
const RevisionHistoryViewer: React.FC<RevisionHistoryViewerProps> = ({ revisions, currentCode, onRestore }) => {
    const [selectedRevisionId, setSelectedRevisionId] = useState<string | null>(null);

    const selectedRevision = revisions.find(r => r.id === selectedRevisionId) || null;

    if (revisions.length === 0) {
        return (
          <div className="flex flex-col items-center justify-center h-full text-text-secondary p-8 text-center">
              <HistoryIcon className="w-16 h-16 mb-4 text-gray-600"/>
              <h3 className="text-2xl font-bold mb-2">No History Yet</h3>
              <p className="max-w-md">Your code snapshots will appear here. Run the script or accept an AI suggestion t
          </div>
        );
    }

    return (
        <div className="h-full flex flex-col lg:flex-row gap-4">
            <div className="lg:w-1/3 flex-shrink-0 flex flex-col">
                <h3 className="text-lg font-semibold mb-3">Code Snapshots</h3>
                <div className="border border-gray-700 rounded-lg overflow-y-auto flex-grow">
                    {revisions.map(revision => (
                        <button
                            key={revision.id}
                            onClick={() => setSelectedRevisionId(revision.id)}
                            className={`w-full text-left p-3 border-b border-gray-700 last:border-b-0 hover:bg-primary/
                        >
                            <p className="font-medium text-sm truncate">{revision.summary}</p>
                            <p className="text-xs text-text-secondary">{timeAgo(revision.timestamp)}</p>
                        </button>
                    ))}
                </div>
            </div>

            <div className="flex-grow flex flex-col">
                 <h3 className="text-lg font-semibold mb-3">Changes</h3>
                 <div className="border border-gray-700 rounded-lg overflow-y-auto flex-grow bg-background/50 p-1">
                    {selectedRevision ? (
                        <div>
                             <DiffViewer
                                oldCode={currentCode}
                                newCode={selectedRevision.code}
                            />
                        </div>
                    ) : (
                        <div className="flex items-center justify-center h-full text-text-secondary">
                            <p>Select a snapshot to see changes.</p>
                        </div>
                    )}
                 </div>
                 {selectedRevision && (
                    <div className="mt-4 flex justify-end">
                        <button
                            onClick={() => onRestore(selectedRevision.code)}
                            className="px-4 py-2 text-sm font-medium text-white bg-secondary rounded-md hover:opacity-8
                        >
                            Restore This Version
                        </button>
                    </div>
                 )}
            </div>
        </div>
    );
};

export default RevisionHistoryViewer;
```

SettingsModal.tsx

```
import React, { useState, useContext, useEffect } from 'react';
import { SettingsContext, Settings, QualitySetting } from '../contexts/SettingsContext';
import { ThemeContext } from '../contexts/ThemeContext';
import { themes } from '../constants/themes';

interface SettingsModalProps {
  isOpen: boolean;
  onClose: () => void;
}

const SettingsModal: React.FC<SettingsModalProps> = ({ isOpen, onClose }) => {
  const { settings, setSettings } = useContext(SettingsContext);
  const { theme, setTheme } = useContext(ThemeContext);

  const [localSettings, setLocalSettings] = useState<Settings>(settings);
  const [selectedTheme, setSelectedTheme] = useState(theme.name);

  useEffect(() => {
    setLocalSettings(settings);
    setSelectedTheme(theme.name);
  }, [settings, theme, isOpen]);

  const handleSave = () => {
    setSettings(localSettings);
    setTheme(selectedTheme);
    onClose();
  };

  const handleCancel = () => {
    onClose();
  };

  if (!isOpen) return null;

  return (
    <div
      className="fixed inset-0 bg-black/60 z-50 flex justify-center items-center"
      onClick={onClose}
    >
      <div
        className="bg-surface rounded-lg shadow-2xl w-full max-w-md border border-gray-700 animate-fade-in text-text-pr
        onClick={(e) => e.stopPropagation()}
      >
        <div className="p-6 border-b border-gray-700">
          <h2 className="text-xl font-bold">Settings</h2>
          <p className="text-sm text-text-secondary mt-1">Configure your AI provider and personalize the UI.</p>
        </div>
        <div className="p-6 space-y-6 max-h-[60vh] overflow-y-auto">
          {/* AI Provider Settings */}
          <div className="space-y-4 p-4 border border-gray-700 rounded-lg">
            <h3 className="font-semibold text-lg text-secondary">AI Configuration</h3>
            <div>
              <label htmlFor="provider" className="block text-sm font-medium text-text-secondary mb-2">
                AI Provider
              </label>
              <select
                id="provider"
                value={localSettings.provider}
                onChange={(e) => setLocalSettings({ ...localSettings, provider: e.target.value as 'gemini' | 'local' })
                className="w-full bg-background border border-gray-600 rounded-md px-3 py-2 focus:ring-2 focus:ring-sec
              >
                <option value="gemini">Google Gemini</option>
                <option value="local">Local AI</option>
              </select>
            </div>
            {localSettings.provider === 'gemini' ? (
              <>
                <div>
                  <label htmlFor="apiKey" className="block text-sm font-medium text-text-secondary mb-2">
                    Gemini API Key
                  </label>
                  <input
```

```
                    type="password"
                    id="apiKey"
                    placeholder="Enter your Gemini API key"
                    value={localSettings.apiKey || ''}
                    onChange={(e) => setLocalSettings({ ...localSettings, apiKey: e.target.value })}
                    className="w-full bg-background border border-gray-600 rounded-md px-3 py-2 focus:ring-2 focus:ring
                  />
                  <p className="text-xs text-text-secondary mt-2">
                    Your key is stored securely in your browser's local storage.
                  </p>
                </div>
                 <div>
                  <label htmlFor="quality" className="block text-sm font-medium text-text-secondary mb-2 mt-4">
                    AI Quality
                  </label>
                  <select
                    id="quality"
                    value={localSettings.qualitySetting}
                    onChange={(e) => setLocalSettings({ ...localSettings, qualitySetting: e.target.value as QualitySett
                    className="w-full bg-background border border-gray-600 rounded-md px-3 py-2 focus:ring-2 focus:ring
                  >
                    <option value="max_quality">Max Quality (Gemini Pro)</option>
                    <option value="balanced">Balanced (Gemini Flash)</option>
                  </select>
                   <p className="text-xs text-text-secondary mt-2">
                    'Balanced' uses a faster model for better interactivity. 'Max Quality' provides more in-depth analy
                  </p>
                </div>
              </>
            ) : (
              <div>
                <label htmlFor="localUrl" className="block text-sm font-medium text-text-secondary mb-2">
                  Local AI URL
                </label>
                <input
                  type="text"
                  id="localUrl"
                  placeholder="http://localhost:5001/api/v1/generate"
                  value={localSettings.localAIUrl || ''}
                  onChange={(e) => setLocalSettings({ ...localSettings, localAIUrl: e.target.value })}
                  className="w-full bg-background border border-gray-600 rounded-md px-3 py-2 focus:ring-2 focus:ring-s
                />
                <p className="text-xs text-text-secondary mt-2">
                  Enter the full API endpoint for your local model.
                </p>
              </div>
            )}
          </div>

          {/* Theme Settings */}
          <div className="space-y-4 p-4 border border-gray-700 rounded-lg">
             <h3 className="font-semibold text-lg text-secondary">Appearance</h3>
             <div>
                <label htmlFor="theme" className="block text-sm font-medium text-text-secondary mb-2">
                    Color Theme
                </label>
                <select
                    id="theme"
                    value={selectedTheme}
                    onChange={(e) => setSelectedTheme(e.target.value)}
                    className="w-full bg-background border border-gray-600 rounded-md px-3 py-2 focus:ring-2 focus:ring
                >
                    {themes.map(t => <option key={t.name} value={t.name}>{t.name}</option>)}
                </select>
             </div>
          </div>
        </div>
        <div className="px-6 py-4 bg-background/50 flex justify-end space-x-3 rounded-b-lg border-t border-gray-700">
          <button
            onClick={handleCancel}
            className="px-4 py-2 text-sm font-medium text-text-secondary rounded-md hover:bg-surface transition"
          >
            Cancel
          </button>
          <button
            onClick={handleSave}
            className="px-4 py-2 text-sm font-medium text-white bg-secondary rounded-md hover:opacity-80 transition"
          >
            Save Settings
          </button>
        </div>
      </div>
       <style>{`
        @keyframes fade-in {
            from { opacity: 0; transform: scale(0.95); }
            to { opacity: 1; transform: scale(1); }
        }
        .animate-fade-in {
            animation: fade-in 0.2s ease-out forwards;
        }
      `}</style>
    </div>
  );
};

export default SettingsModal;
```

Sidebar.css