Files and Their Functions

1. Main.html
   a. This file creates the user interface for the extension's popup.
   b. It includes styling for the popup, such as font, color, and button appearance.
   c. Contains an input field for the OpenAI API key and a save button.
   d. The popup is designed with a clean, user-friendly interface.
2. Popup.js
   a. Handles the logic for the popup interface.
   b. Retrieves and stores the OpenAI API key in Chrome's local storage.
   c. Displays a success message upon saving the API key, or prompts for a valid key if the field is empty.
   d. Closes the popup automatically after successfully saving the API key.
3. Content.js
   a. Injected into all web pages (as per the manifest file).
   b. Listens for a mouseup event to capture selected text.
   c. Sends the selected text to the background script for processing.
4. Background.js
   a. The core script that interacts with the OpenAI API.
   b. Receives text from the content script and fetches insights using the OpenAI API.
   c. Handles API requests and responses, including error handling.
   d. Sends the received insights back to the content script.
5. Manifest.json
   a. The configuration file for the Chrome extension.
   b. Sets the manifest version, name, version, and description of the extension.
   c. Defines the extension's popup, background scripts, content scripts, permissions, and host permissions.
   d. Includes icons for the extension in various sizes.

Key Features

- API Key Configuration: Users can enter and save their OpenAI API key through the extension's popup interface.
- Text Selection and Processing: Users can select text on any webpage, which is then sent to OpenAI for contextual insights.
- Asynchronous Communication: The extension uses message passing between the content and background scripts for asynchronous operations.

- Error Handling: The extension includes basic error handling, particularly in the interaction with the OpenAI API.

Installation and Usage

- Installation: To install the extension, users should load the unpacked extension into Chrome through the extensions page (chrome://extensions/).
- API Key Setup: Users need to enter their OpenAI API key in the popup and save it for the extension to function.
- Using the Extension: Simply highlight text on any webpage, and the extension will provide contextual insights using OpenAI's API.

Report on Deviation from Original Project Plan to Implementation

This report outlines the deviations from the initial project proposal and progress report to the final implementation of the "Contextual Insights" Chrome extension. Originally, the project was intended to be a browser extension that provided contextual insights using user-generated annotations, articles, and NLP techniques. The final product, however, utilized the OpenAI API for its core functionality.

Deviation and Implementation

- Shift to OpenAI API: The project took a significant turn by adopting the OpenAI API. This decision was driven by the challenges faced in developing and integrating efficient NLP techniques and creating a user-friendly interface.
- Rationale for Change:
  - Integration Challenges: Difficulties in integrating NLP techniques and optimizing performance were significant hurdles.
  - Efficiency and Accuracy: The OpenAI API offered a more robust and accurate solution for generating contextual insights.
  - User Experience: The complexities in designing an intuitive UI that could handle the planned features efficiently led to a shift towards a more streamlined solution.