Task 8-Theme (Summary)

* **Gemini’s Approach**:
  + It defines separate variables and uses a conditional mixin, which can become limiting if more themes are added or if the logic becomes more complex.
  + Sass is **compiled**, so managing runtime state (like toggling between light and dark themes) in compiled code can be tricky without integrating CSS variables.
* **ChatGPT’s Approach**:
  + Using maps and a mixin-based theme system allows for greater scalability. If Gemini want to add more themes (like a "blue mode" or "high-contrast" mode), it would be easier by simply adding new maps and using classes to switch themes dynamically.
  + ;ChatGPT’s approach leverages JavaScript more effectively, where the theme state (which theme is active) is managed entirely through the DOM (by adding/removing classes).

**Conclusion**

* **Gemini’s approach** is straightforward but limited in its scalability. It attempts to toggle the theme within Sass, which won't work as Sass is a preprocessor and doesn’t handle runtime logic well.
* **ChatGPT approach** takes advantage of Sass maps for organization and JavaScript for runtime theme switching, making it more maintainable, scalable, and efficient for larger applications with more themes and styles.