**AI Tool Declaration**

In the course of developing the web application on Java Spring Boot and ReactJS, Team 4 has used AI tools to perform the following tasks:

1. Come up with the brand concept for Aori.
2. Propose a suitable database schema for our envisioned fashion e-commerce platform.
3. Generate and refine ideas of how the sequence diagrams for the respective use cases the team had selected to work on.
4. Generation of test data, including all the images used.
5. Generate the first cut of the code base for the Aori application on Java Spring Boot, React JS and Thymeleaf.
6. Refined the codes further after generating initial versions to adapt it to our current context and database design.
7. Use of in-built code-assist tools to troubleshoot and debug issues with the code, especially with regards to integration.

The tools used by the team are listed here:

Anthropic. (2025). *Claude 4 Sonnet* [Large language model]. <https://claude.ai/new>

Google. (2025). *Gemini 2.5 Flash* [Large language model]. [https://gemini.google.com](https://gemini.google.com/)

Google. (2025). *Gemini 2.5 Pro* [Large language model]. [https://gemini.google.com](https://gemini.google.com/)

Microsoft. (2025). Copilot [Large language model]. <https://copilot.microsoft.com/>

OpenAI. (2025). *ChatGPT4* [Large language model]. <https://chatgpt.com/>

OpenAI. (2025). *ChatGPT5* [Large language model]. <https://chatgpt.com/>

In terms of website design inspiration, we have used this as our base template to design and refine our website design on ReactJS.

Hakimi, Y. (2024). *Cloth Store / Fashion Store E-Commerce UI Kit* [Graphic user interface mock-up]. Figma. <https://www.figma.com/community/file/1365263619600313207/cloth-store-fashion-store-e-commerce-ui-kit>

**Learning Points**

Our team's intent was to come up with a working Minimum Viable Product that we could showcase for our fashion e-commerce website. While we have used AI tools extensively in the process of building the application, the team would like to share that we have learnt a lot from this process of working together.

1. **Standardisation of naming of service layers**
   1. Interfaces are called prefixed with a capital "I".
   2. Service implementation are generally called Service.
2. **Creation of common services that can be used across various use cases or functions, such as:** 
   1. Interceptor for authentication
   2. Utils for authentication, get Session and Sku tool
   3. Config for interaction with Thymeleaf, ReactJS and even for debugging purposes
3. **Thymeleaf:** Leveraging on a common layout template to standardize how the various pages post-login should look like.