

AI Usage Report – Using Gemini via Vibe Coding

This report documents how the Gemini AI tool was utilized through Vibe Coding during the development of a React Native project. The project aimed to develop a fully functional product listing app that displays data from a public API, enables search and filtering, manages favorite items with local persistence, and includes detailed product screens.

1. Tools Used and Why (5 Marks)

I used **Gemini (via Vibe Coding)** to assist with React Native code generation, optimization, and UI improvement. Gemini helped create well-structured functional components, implement efficient state handling using React Hooks, and suggested responsive layouts using Flexbox.

2. Actual Prompts and Outputs (10 Marks)

Prompt Example 1:

"Generate a React Native FlatList to display product cards using data from <https://fakestoreapi.com/>"

Gemini's Output:

Gemini generated a component using *fetch()* API for data retrieval and displayed the products in a FlatList. Each product card included an image, title, and price with basic styling.

Prompt Example 2:

"Add a search bar to filter products by name and category."

Gemini's Output:

Gemini created an efficient filtering mechanism using *useState* and *useEffect* to dynamically filter results based on search input.

3. How I Improved, Modified, or Verified AI-generated Code (10 Marks)

I modified Gemini's code to improve reusability by separating UI components such as ProductCard and SearchBar. I also added local storage functionality using *AsyncStorage* for favorites persistence. Testing was performed on both Android and iOS devices to verify the app's layout, functionality, and responsiveness.

4. Screenshot Evidence (5 Marks Bonus)

Below is the actual screenshot showing Gemini's assistance within Vibe Coding, where prompts and AI-generated code responses were used during the project.

