Answers to common interview questions:

1. **Question: “Walk me through one UI Bart has built end-to-end that he is proud of — what problem it solved, the tech stack used, and the key tradeoffs made.”**
   (Tests practical experience, ownership, and product thinking.)

   Answer: While at Grid Dynamics, Bart worked on re-platforming a major U.S. bank’s public website and designed a reusable, searchable multi-select account filter component for the navbar. Originally, the product relied on a Material UI single-select dropdown, which was painful for customers with dozens or hundreds of accounts. Bart built a React component that accepted a tree data structure, letting users expand and collapse account groups and search by keyword without leaving the navbar. He designed it as a shared component, so other teams could easily integrate it across the application, and it quickly saw adoption on multiple pages. To handle differing interaction expectations, Bart exposed flexible configuration props (for example, how to display existing selections during search) without breaking existing implementations. For clients with very large account lists, he collaborated with design and product to switch from a dropdown to a drawer when account counts exceeded a threshold, improving usability and performance for those users.

2. **Question: “How does Bart approach building UIs that are fast, accessible, and work well across different devices and browsers?”**
   (Surfaces their mental model for performance, a11y, and responsive design, not just buzzwords.)

   Answer: Bart has built dozens of websites and web applications with an emphasis on speed, accessibility, and consistent behavior across devices and browsers. His process starts with mobile-first design, responsive layouts, and a clear visual hierarchy so key actions are always obvious. He is an advocate of redundant navigation and using visual cues to guide interaction, recognizing that users have different experience levels and expectations. Bart builds iteratively, regularly reviewing and testing UIs on multiple devices and browsers early in development to catch layout and interaction issues before they become expensive to fix. Influenced by Steve Krug’s “Don’t Make Me Think,” he focuses on reducing cognitive friction so interfaces feel intuitive, require minimal explanation, and are enjoyable to use.

3. **“Tell me about a time Bart and a designer or product manager strongly disagreed about a UI decision — what happened and how was it resolved?”**
   (Probes collaboration, communication, and conflict resolution in a real product environment.)

   Answer: While at Grid Dynamics, Bart’s team was building a new drawer component that let users configure filters and save them as reusable “named searches.” During

refinement, Bart realized that choices in some filters would need to dynamically change the options available in others, which required additional logic and API calls that were not accounted for in the existing user stories or schedule. When he raised this dependency, the product managers initially preferred to stay with the original plan and timeline. Bart persisted, clearly outlining the user experience risks and technical implications of ignoring the interdependencies, and walked through concrete scenarios to illustrate potential failures. Product ultimately agreed to adjust the roadmap, extend the schedule, and rework the user stories, resulting in a more coherent, maintainable implementation of the feature.

4. **“In Bart’s view, what distinguishes a good UI engineer from a great one on a modern front-end stack?”**
   (Reveals their standards, how they see the role, and whether they align with a high bar for quality and impact.)

Answer: Bart believes a good UI engineer can work effectively within an existing codebase to implement features, fix bugs, and extend components. A great UI engineer goes further by designing clear abstractions for data structures, components, and data flow, so the system remains scalable and maintainable as it grows. They think in terms of architecture—component boundaries, state management, and API contracts—rather than only individual screens. Great UI engineers also collaborate closely with design and product to ensure that technical decisions support a consistent, high-quality user experience over time.