Question: "Describe a project that Bart is particularly proud of — what were the challenges, solutions, and tradeoffs?",

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.

Question: "How does Bart make UIs fast, accessible, and cross-device?",

Answer: Bart follows an agile development process. Working with an iterative approach, he develops an MVP (minimal viable product), first, and then expands on it incrementally. He always reviews work in progress on multiple browsers and devices and screen sizes. He starts with mobile first, and keeps designs flexible and responsive. His approach is to keep API responses small to increase responsiveness, and works with back-end developers to design APIs and data contracts to accomplish this. He uses developer tools and benchmarking to verify rendering speeds and API response timing. While working at Grid Dynamics, he developed a lazy-loading API request and response mechanism that was designed to read the first 100 records of a clients transaction history at a time, and then as the user scrolls, it preloads the next 100 records, so the user experiences smooth scrolling through thousands of records.

Question: "Share a time Bart and a designer or PM disagreed — what happened and outcome?",
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.

Question: "Would Bart be a good for a Series B startup with messy data infrastructure?",

Answer: Yes. Bart can confidently handle data mapping and data restructuring tasks. In a previous career he also worked for several Start Ups. He has worked for numerous ad agencies and web development shops , working on tight deadlines. He did web development for a 6 person advertising and public relations firm, McQuerter Group, based in San Diego CA. He also worked for Platinion which was a start-up venture of Boston Consulting Group, where he was the Design Director and had a staff of 4 web designers working on creative, innovative projects.

Question: "Tell me about a time Bart chose a technology platform that he later changed his mind about.",

Answer: A few years ago, Bart made a blunder choosing a technology platform. He chose Angular to build a large-scale Learning Management System (LMS) . But in hindsight he wishes he had taken the time to experiment with it on a smaller scale, first.
At the time, Angular seemed like the right choice. It was popular, well-documented, and backed by Google. He dove headfirst into building a full-featured LMS with user dashboards, course management, and analytics. But as the project grew, so did his frustration. He kept running into architectural challenges, performance issues, and a steep learning curve that made even simple features take longer than expected.
In hindsight, He realized he skipped a crucial step — validation. He should have started with a small "proof-of-concept" project to understand Angular's strengths and limitations. If he had done that, he would've quickly recognized that React would have been a better fit for the team's skill set and the application's requirements.
That experience taught him an enduring lesson: before committing to a new technology for a major build, always test it on a smaller project first. A few days or weeks of exploration can save months of pain down the line.

"**Question**: What kind of leadership experience does Bart have?",
Answer: Bart is a natural mentor. He enjoys teaching other developers and regularly reviews fellow engineers code, giving constructive feedback. He always keeps the long term scalability and readability of the codebase in mind. At Grid Dynamics he was the lead for an agile team of four front-end developers.

Question: "How does Bart approach UI/UX design and development?",
Answer: Bart studied art and design at the prestigious Pratt Institute. He has a previous career in advertising and 3D animation, and 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.