Integrating Three.js with Shopify for Immersive Storefront Experiences (2024-2025)

I. Introduction: The Rise of Immersive E-commerce with Three.js and Shopify

The landscape of online retail is undergoing a significant transformation, driven by evolving consumer expectations and advancements in web technologies. Contemporary shoppers increasingly seek more than just static images and textual descriptions when browsing products online. They desire engaging and interactive experiences that closely mimic the tactile and visual aspects of physical shopping.[1] This demand has fueled the rise of immersive e-commerce, where technologies like 3D visualization play a pivotal role in enhancing customer engagement and ultimately boosting conversion rates.[1] The ability to interact with products in a three-dimensional space provides a richer understanding of their form, features, and origin, fostering greater confidence in purchasing decisions.[1]

At the forefront of enabling these immersive experiences on the web is Three.js, a versatile and powerful cross-browser JavaScript library. Built upon the WebGL API, Three.js allows developers to create and display animated 3D computer graphics directly within web browsers.[3] Its comprehensive set of features and inherent flexibility make it an ideal tool for crafting intricate and interactive 3D environments without requiring users to install additional plugins.[5] From detailed product showcases to engaging brand narratives, Three.js empowers developers to push the boundaries of online visual communication.

Shopify, a leading e-commerce platform, provides businesses with a robust infrastructure to establish and manage their online stores. While Shopify offers a wide array of themes and built-in functionalities, the integration of highly customized 3D experiences often requires solutions that extend beyond these standard offerings.[7] The platform's architecture, while powerful for conventional e-commerce, may not inherently support the nuanced requirements of complex, interactive 3D scenes. To fully leverage the potential of immersive technologies like Three.js, developers often need to explore custom integration strategies that provide granular control over design and functionality.[7] By incorporating custom JavaScript frameworks, businesses can create truly unique and memorable shopping experiences that differentiate their brand and captivate their audience.

This report serves as a comprehensive guide for developers and technical decision-makers seeking to integrate Three.js-powered experiences into Shopify storefronts during the years 2024 and 2025. It will delve into various approaches for

achieving this integration, provide recommendations on suitable JavaScript frameworks, detail strategies for ensuring hydration-safe rendering of 3D content, highlight valuable example repositories and learning resources, and offer best practices for designing and implementing a polished and performant splash page that incorporates a Three.js experience with deep metafield integration. The objective is to equip technical teams with the knowledge and insights necessary to create compelling and effective immersive e-commerce solutions on the Shopify platform.

II. Foundational Approaches to Three.js Integration in Shopify (2024-2025)

For developers aiming to incorporate Three.js into a Shopify storefront, several fundamental approaches can be considered, each offering a distinct set of capabilities and complexities. The most direct method involves integrating custom JavaScript code within the existing Shopify theme, which is built using the Liquid templating language.[2] This can be achieved by directly modifying theme files such as theme.liquid for global scripts or specific template files like product-template.liquid for page-specific functionality. Additionally, the Assets folder within the theme provides a location to upload external JavaScript files, which can then be linked within the theme's templates using Shopify's asset_url tag.[11] This method offers a relatively straightforward entry point for developers familiar with Shopify's theme structure and Liquid. However, for more intricate 3D scenes and long-term maintainability, this approach can become less manageable.

A more robust and flexible strategy involves leveraging Shopify's Storefront API to build completely custom online stores, often referred to as headless commerce.[7] This architecture separates the frontend of the store from Shopify's backend infrastructure, granting developers full control over the design and functionality of the customer-facing experience. Frameworks such as Next.js, Vue.js, or Svelte can be employed to build these custom frontends, which then communicate with Shopify's backend via the Storefront API.[7] Implementing this approach necessitates setting up a Shopify account with administrative access, installing the Headless channel app, and generating API access tokens to facilitate secure communication between the custom frontend and Shopify's data.[7] While requiring a more significant initial development effort, headless commerce offers unparalleled flexibility and performance for highly customized 3D experiences.

Another viable approach involves embedding Three.js directly within existing Shopify themes using standard <script> tags that link to Content Delivery Network (CDN)-hosted versions of the Three.js library.[3] This method allows for a rapid integration of Three.js for simpler 3D elements or product viewers without the need

for complex build processes or extensive framework integrations. By adding a few lines of HTML to a theme template, developers can immediately access the functionalities of Three.js. However, this approach might become less organized and harder to maintain for more complex 3D scenes that require a structured framework.

Finally, Shopify Apps present an alternative method for injecting custom code, including JavaScript, into a Shopify store.[16] These apps can be designed to add custom HTML, CSS, and JavaScript to enhance a store's design and functionality without requiring direct modifications to the theme files. This approach can simplify the integration process and offer a more managed environment for custom code, potentially reducing the risk of conflicts during theme updates. By utilizing an app, developers can often benefit from a dedicated interface for managing and deploying their custom Three.js integrations.

III. Deep Dive: Integrating Three.js with Vue.js and TroisJS for Shopify

Vue.js has emerged as a popular progressive JavaScript framework, celebrated for its user-friendliness, adaptability, and rapid development capabilities.[7] Its component-based architecture fosters the creation of reusable and easily maintainable code structures [22], which is particularly advantageous for developing interactive storefronts with dynamic elements. Vue.js excels at building engaging user interfaces that can update content in real-time without requiring full page reloads [2], contributing to a smoother and more responsive shopping experience. The framework's compatibility with the Shopify Storefront API further solidifies its position as a strong contender for building custom e-commerce experiences.[7]

For Vue.js developers looking to integrate Three.js into their Shopify projects, TroisJS offers a compelling solution. TroisJS acts as a Vue layer for Three.js, providing a set of Vue components that directly correspond to Three.js objects.[22] This abstraction allows developers to build 3D scenes using Vue's familiar declarative syntax, making the process more intuitive for those already comfortable with Vue.[22] Several users within the Shopify community have reported success using TroisJS, even in scenarios where direct import of newer Three.js versions encountered issues.[25] Furthermore, TroisJS is being actively used with modern Vue.js tooling like Vue 3 and Vite [28], indicating its continued relevance in the Vue.js ecosystem.

When employing Vue.js and TroisJS within a Shopify environment, adhering to certain best practices can ensure a smooth and efficient integration. Adopting a component-based approach, where Three.js elements are encapsulated within Vue components, promotes better code organization and enhances reusability.[22] Defining

the 3D scene structure using TroisJS components within Vue templates allows for a declarative and easily understandable representation of the 3D environment.[23] Leveraging Vue's reactivity system enables dynamic updates and animations of Three.js objects based on changes in Vue's data [28], creating interactive and engaging experiences. Integrating these scripts within the Shopify theme structure typically involves including the necessary Vue.js and TroisJS scripts in the theme's layout file (theme.liquid) or within specific template files.[31] Developers can choose between using CDN links for simpler setups or opting for local installation and build processes for more complex applications or offline development.[20]

Despite its advantages, there are potential challenges to consider when using Vue.js and TroisJS in Shopify. As noted by some sources, TroisJS might not be as actively maintained as other libraries, which could lead to long-term compatibility issues with newer Three.js versions.[22] Performance optimization is also crucial, as complex 3D scenes can impact page load times. Developers should implement optimization techniques specific to both Three.js and Vue.js, such as lazy loading and component optimization.[25] If server-side rendering is employed with Vue.js, ensuring proper hydration of the Three.js/TroisJS elements is vital to avoid client-side rendering discrepancies.[33]

IV. Exploring the Power of React and Shopify Hydrogen for Three.js Experiences

React stands as a leading JavaScript library for crafting intricate user interfaces, renowned for its high performance, component-driven architecture, and a vast and supportive ecosystem.[6] Its utilization of a virtual DOM significantly enhances performance by enabling efficient updates to the user interface [37], making it well-suited for building complex and dynamic storefronts that incorporate 3D elements. The extensive adoption of React and its vibrant community ensure a wealth of resources and readily available support for developers.[18] Frameworks like Next.js and Shopify Hydrogen further extend React's capabilities for building robust e-commerce solutions, seamlessly integrating with Shopify's Storefront API.[7]

Shopify has specifically developed Hydrogen as a React-based framework tailored for building high-performing, custom headless storefronts.[34] Hydrogen provides developers with a collection of Shopify-specific React components, hooks, and utilities designed to streamline interactions with the Storefront API.[34] Leveraging modern React features such as Server Components and streaming server-side rendering, Hydrogen prioritizes fast initial load times and overall performance.[36] The Hydrogen React library, which contains the React-specific functionalities for interacting with the Storefront API, can also be used independently of the full

Hydrogen framework.[34] Built upon the Remix framework [45], Hydrogen offers a structured and efficient approach to developing complex e-commerce experiences.

Integrating Three.js within a Hydrogen project is achievable by incorporating it within React components.[54] Libraries like React Three Fiber (@react-three/fiber) significantly simplify this process by providing a declarative API for managing Three.js scenes and objects within a React environment.[5] This allows React developers to work with Three.js in a manner consistent with their existing skill set and familiar paradigms. Several examples and community discussions highlight the successful integration of React Three Fiber with Hydrogen for creating interactive 3D elements, including full 3D product carousels.[54]

Utilizing Hydrogen for Three.js integration offers several advantages. The inherent performance benefits of server-side rendering and React Server Components in Hydrogen contribute to faster load times and a more responsive user experience.[36] The framework's deep integration with Shopify's features and APIs simplifies the development of e-commerce-specific functionalities. Furthermore, Hydrogen provides a structured framework that is well-suited for building complex and scalable storefronts.[36] However, it's important to acknowledge the potential disadvantages. Hydrogen has a steeper learning curve compared to direct theme integration and requires a more involved setup process. Additionally, it necessitates a reliance on the React ecosystem, which might not be ideal for developers unfamiliar with React.[46]

V. Framework Recommendations for Interactive 3D on Shopify: A Comparative Analysis

When considering the optimal framework for integrating interactive 3D experiences into a Shopify storefront, several factors come into play, including team expertise, project complexity, performance requirements, and long-term maintainability.

For development teams already proficient in Vue.js, the combination of Vue.js and TroisJS presents a viable option. This stack is particularly suitable for projects requiring rapid development and for embedding interactive 3D elements within existing Shopify themes. Vue.js offers a relatively gentle learning curve, and TroisJS provides a declarative way for Vue developers to define 3D scenes, potentially leading to quicker integration. However, the long-term maintenance and update status of TroisJS should be carefully considered, and complex 3D scenes might necessitate more manual performance optimization compared to other frameworks.

For more ambitious projects that prioritize performance, scalability, and deep

integration with Shopify's headless commerce features, the React + Hydrogen + React Three Fiber stack emerges as a strong recommendation. This combination is ideal for building completely custom, 3D-centric storefronts. Hydrogen's architecture, leveraging server-side rendering and React Server Components, delivers excellent performance. Access to Shopify's headless commerce functionalities, coupled with the mature and actively maintained React Three Fiber library, makes this a powerful and future-proof solution. While the initial learning curve and setup complexity are higher, the benefits in terms of performance and control are often worthwhile for complex endeavors.

Direct Three.js integration within existing Shopify themes offers a lightweight solution for very simple 3D elements or product viewers that do not require the full structure of a framework. This approach is suitable for quick enhancements to existing themes without significant architectural changes. It provides direct control over the Three.js implementation with minimal overhead. However, managing complex 3D scenes can become challenging, and achieving optimal performance might require meticulous manual optimization. This method lacks the structured approach and inherent maintainability benefits offered by dedicated frameworks.

Ultimately, the selection of the most appropriate framework hinges on a careful evaluation of the project's specific needs and constraints. Teams should leverage their existing expertise to minimize the learning curve. The complexity of the desired 3D experience should guide the choice towards more robust frameworks for intricate scenes. Performance-critical applications will likely benefit from the performance-focused architecture of Hydrogen. Finally, considering the long-term maintainability of the chosen technologies is crucial for the project's sustained success. The level of integration desired – whether a deep, custom storefront or specific enhancements within an existing theme – will also play a significant role in the decision-making process.

VI. Achieving Seamlessness: Hydration-Safe Rendering of Three.js Content within Shopify

To deliver a smooth and performant user experience, particularly when integrating interactive 3D content, understanding the nuances of rendering strategies is paramount. Client-Side Rendering (CSR) and Server-Side Rendering (SSR) represent two fundamental approaches to how web applications are rendered and become interactive. In CSR, the browser initially receives a minimal HTML page. It then fetches and executes JavaScript code to dynamically generate and display the main content.[60] While this approach can lead to a rich and interactive user experience after the initial

load, it can also result in noticeable delays in content visibility, especially on slower network connections or when dealing with large JavaScript bundles.[60]

Conversely, SSR involves the server pre-rendering the initial HTML of the page and sending this fully formed HTML to the browser.[60] This significantly improves the initial load time, as the user sees content much faster. However, the HTML received from the server is initially static and lacks interactivity. To bridge this gap, a process known as hydration is essential. Hydration is the mechanism by which client-side JavaScript takes over the server-rendered static HTML, attaching event listeners and restoring the application's state to make it fully interactive.[60]

For Three.js experiences within a Shopify environment, particularly when using SSR frameworks like Hydrogen, ensuring hydration-safe rendering is crucial. This involves several key strategies. One effective technique is to render an empty container element, such as a or <canvas>, on the server as a placeholder for the Three.js scene.[71] The actual initialization of the Three.js renderer, scene, and camera should then be performed exclusively on the client-side within the appropriate lifecycle hooks, such as useEffect in React or onMounted in Vue.js.[71] This ensures that the Three.js code, which relies on browser-specific APIs, only runs in the browser environment, preventing potential errors during server-side rendering.

Maintaining a consistent DOM structure between the server-rendered output and the client-side Three.js setup is also vital to avoid hydration mismatches.[64] Discrepancies in the HTML structure can lead to errors during the hydration process, causing the client-side application to potentially discard the server-rendered content and re-render from scratch, negating the performance benefits of SSR. Developers should also exercise caution when using browser-only APIs like window or document within their Three.js initialization code, ensuring these are accessed only after the component has mounted on the client.[72] In situations where minor, unavoidable differences between server and client rendering might occur (e.g., dynamic timestamps or unique IDs), frameworks often provide mechanisms to suppress hydration warnings on specific elements, such as React's suppressHydrationWarning.[64]

To further optimize performance, especially for complex Three.js scenes, consider implementing lazy loading for the Three.js library and associated assets.[67] This technique ensures that these resources are only loaded when they are about to become visible or are needed, minimizing the initial page load time. For Vue.js applications using TroisJS with SSR, the same principles apply: render a placeholder server-side and initialize the TroisJS components within the onMounted hook.

Consistent data fetching and management across both server and client are also essential.

When working with React and Shopify Hydrogen, the framework's architecture, which incorporates React Server Components and streaming SSR, can inherently aid in achieving better initial rendering performance.[47] Developers should ensure that the React Three Fiber canvas and its contents are initialized correctly on the client after the hydration process is complete.[47] React's Suspense component can be effectively utilized to manage the asynchronous loading of Three.js components and assets during hydration, providing a fallback UI while the 3D content is being loaded.[54] Modern React versions also offer features like selective hydration, where React intelligently prioritizes the hydration of interactive elements, further improving the perceived performance and time to interactivity.[67]

VII. Unlocking Dynamic Content: Integrating Shopify Metafields with Three.js Scenes

Shopify metafields provide a flexible and powerful mechanism for storing additional custom data that extends beyond the standard fields available for Shopify resources like products, collections, customers, and orders.[45] Each metafield comprises a namespace, a key, a value, and a data type.[77] These metafields can be defined and managed directly within the Shopify admin interface or programmatically through the Shopify Admin API.[77] This capability allows merchants to associate rich, structured data with their store's entities, enabling highly customized storefront experiences.

To leverage this custom data within a Three.js-powered storefront, developers can access Shopify metafields using the platform's Storefront API.[7] Retrieving metafield data typically involves constructing GraphQL queries to the Storefront API, specifying the namespace and key of the desired metafield.[81] It is important to note that for metafields created via the Shopify admin to be accessible through the Storefront API, their storefront visibility must be explicitly enabled using the metafieldStorefrontVisibilityCreate mutation in the Admin API.[80] This two-step process ensures that sensitive or internal-only metafield data is not inadvertently exposed on the public storefront.

Once the metafield data is successfully retrieved within the JavaScript code of the storefront, it can be seamlessly integrated into the Three.js scene to dynamically control various aspects of the 3D environment.[86] The possibilities for this integration are vast and can significantly enhance the interactivity and personalization of the 3D experience. For instance, metafield values can be used to alter the color or material of

3D objects based on a color metafield, adjust the size or scale of elements according to numerical metafields, load different 3D models based on a metafield referencing a file, display dynamic text labels or annotations using data from text metafields, or even control animations and interactive features based on specific metafield values.

To ensure a robust and maintainable integration of metafields with Three.js, several best practices should be followed. Developers should carefully define the structure of their metafields, including the namespace, keys, and data types, to align with the intended dynamic controls within the 3D scene. Optimizing Storefront API queries to fetch only the necessary metafields will help minimize data transfer and improve overall performance. When metafield data changes, updating only the relevant parts of the Three.js scene will prevent unnecessary re-renders, further enhancing performance. It is also crucial to consider potential data type conversions, ensuring that the metafield value's type matches what is expected by the Three.js code (e.g., converting string values to numbers or color objects). Implementing fallback mechanisms or default values will provide a more resilient experience in cases where metafield data might be missing or invalid.

In the context of Vue.js and TroisJS, metafield data can be retrieved within Vue components using the Storefront API. Vue's powerful data binding capabilities can then be used to pass these metafield values as props to the TroisJS components, dynamically controlling their properties and appearance. For React-based storefronts using Hydrogen and React Three Fiber, metafield data can be fetched using Hydrogen's data fetching mechanisms, such as loaders. This data can then be passed as props to the React components that contain the React Three Fiber canvas. React's state management can be employed to control the properties of Three.js elements based on the received metafield props. Hydrogen also provides specific utilities, such as the parseMetafield function, which aids in parsing metafield values based on their defined data type.[90] This simplifies the process of working with different types of metafield data within a Hydrogen environment.[92]

VIII. Practical Guidance: Example Repositories and Learning Resources for Three.js and Shopify

For developers embarking on the journey of integrating Three.js with Shopify, a wealth of practical guidance and example code is available across various online platforms. The official Three.js website hosts an extensive collection of examples that showcase the library's diverse features and techniques.[93] These examples serve as invaluable resources for understanding specific Three.js functionalities relevant to e-commerce applications, such as 3D model loading, material properties, lighting techniques, and

user interaction implementations.[93]

Developers opting for the Vue.js and TroisJS approach can find relevant examples within the official TroisJS documentation and its corresponding GitHub repository.[29] These resources often include demonstrations of how to use TroisJS components to build 3D scenes within a Vue.js application.[28] Additionally, numerous online tutorials and articles provide practical examples and step-by-step guides for integrating these technologies.

For those leveraging React and Shopify Hydrogen, the React Three Fiber documentation offers a wide range of examples illustrating how to effectively use Three.js within a React environment.[58] Shopify's Hydrogen documentation and its example storefront projects may also contain valuable insights into integrating custom React components, which can be adapted for incorporating Three.js using React Three Fiber.[94] Furthermore, community-driven repositories on platforms like GitHub often showcase specific integrations of React Three Fiber with Hydrogen, providing real-world examples and potential starting points for projects.[55]

Beyond framework-specific examples, searching GitHub directly for "shopify three.js" can uncover relevant open-source projects, code snippets, and potential solutions shared by the developer community.[30] The Shopify developer community forums and other online forums dedicated to Three.js and web development serve as invaluable resources for finding examples, troubleshooting issues, and discovering best practices from fellow developers.[25]

For efficient management of custom code, including Three.js integrations within Shopify themes, integrating the theme with a version control system like GitHub is highly recommended.[97] This allows for seamless synchronization of changes between local development environments and the Shopify admin, facilitating collaboration and providing a robust mechanism for tracking and managing code modifications.[97]

Finally, the official developer documentation provided by both Shopify [9] and the respective JavaScript frameworks (Vue.js [20] and React) offer comprehensive information on their features, APIs, and best practices. These documentation resources are essential for gaining a deep understanding of the technologies involved and for effectively implementing Three.js integrations within the Shopify ecosystem.

IX. Crafting the First Impression: Best Practices for a Performant 3D Shopify Splash Page

A splash page serves as the initial point of contact for visitors to a Shopify store,

offering an opportunity to create a memorable first impression. These pages can be strategically employed to greet visitors, convey important information, highlight special promotions, or even implement age verification measures.[99] When incorporating a 3D experience, the splash page can evolve into an engaging introduction to the brand, showcasing interactive elements that immediately captivate the user's attention.[100]

To create an effective 3D Shopify splash page, several key elements should be considered. High-quality visuals are paramount; investing in compelling and brand-consistent 3D graphics will ensure a positive initial perception.[99] The messaging on the splash page should be clear and concise, quickly conveying the core value proposition or the key message the brand wishes to communicate.[99] A strong call to action (CTA) is essential to guide users on the next step, whether it's "Enter Store," "Explore 3D Products," or another relevant action.[100] Providing a clear and seamless entry point to the main store is crucial; users should easily be able to exit the splash page and navigate to the rest of the website.[99] Given the prevalence of mobile browsing, optimizing the splash page for mobile devices is non-negotiable, ensuring a responsive and functional experience across all screen sizes.[99] Finally, and perhaps most importantly for a 3D experience, performance optimization is critical. Minimizing loading times for the 3D elements will help prevent high bounce rates and ensure that users are not discouraged by lengthy loading screens.[102]

Optimizing the performance of a Three.js-powered splash page involves a multifaceted approach. Model optimization is key; utilizing low-poly models, compressing textures, and carefully optimizing materials can significantly reduce file sizes and the rendering load on the user's device.[1] Texture optimization techniques, such as using compressed formats like WebP, limiting texture resolution, and reusing textures wherever feasible, will further contribute to faster loading and smoother rendering.[103] Code optimization is equally important; minimizing JavaScript usage, avoiding parser-blocking scripts, and optimizing the rendering loop will enhance performance.[107] Implementing appropriate loading strategies, such as lazy loading for non-critical assets and providing loading screens or progress indicators for the 3D scene, can improve the perceived performance.[102] For scenes with multiple levels of detail, employing Level of Detail (LOD) techniques, where different versions of 3D models with varying detail are displayed based on the distance from the camera, can significantly boost rendering efficiency.[103] If the scene includes multiple identical 3D objects, using instancing to render them in a single draw call can dramatically reduce the overhead.[105]

Integrating the 3D experience into the splash page involves embedding the Three.js

canvas within the page's HTML structure. Ensuring that the canvas is appropriately sized and positioned to adapt to different screen dimensions is crucial for a seamless visual experience. The loading and initialization of the 3D scene should be carefully managed to optimize the initial viewing, potentially delaying the start of complex animations or resource-intensive processes until the necessary assets are fully loaded.

Deep metafield integration can add a powerful layer of dynamic content and personalization to the 3D splash page. Metafields can be used to control various aspects of the splash page's content and appearance. For example, a metafield could specify the particular 3D model to be loaded, the color scheme to be applied to the scene, or promotional text to be displayed within the 3D environment. By retrieving these metafield values and using them to configure the Three.js scene accordingly, developers can create highly tailored and context-aware splash page experiences that resonate with individual visitors.

X. Conclusion: Future-Proofing Your Shopify Store with Immersive 3D

Integrating Three.js-powered experiences into Shopify storefronts presents a significant opportunity for businesses to elevate customer engagement and drive conversions in the increasingly competitive online marketplace. This report has explored various foundational approaches, ranging from direct theme integration to leveraging the full potential of headless commerce with Shopify's Storefront API and frameworks like Vue.js with TroisJS and React with Hydrogen. Each approach offers unique advantages and considerations regarding complexity, performance, and maintainability.

The choice of framework ultimately depends on the specific requirements of the project, the expertise of the development team, and the desired level of integration. Vue.js and TroisJS provide a more accessible path for Vue developers, while React and Hydrogen offer a robust and performant solution for complex, high-performance 3D storefronts. Direct Three.js integration remains a viable option for simpler enhancements. Regardless of the chosen path, ensuring hydration-safe rendering is crucial for delivering a seamless user experience, especially when utilizing server-side rendering techniques.

Shopify metafields emerge as a powerful tool for unlocking dynamic content within these 3D experiences, allowing for deep customization and personalization based on product data and other store resources. By carefully planning the metafield structure and utilizing the Storefront API, developers can create truly data-driven and

interactive 3D storefronts.

The wealth of learning resources and example repositories available for Three.js, Vue.js, React, and Shopify provides a solid foundation for developers to build upon. The active and supportive developer communities surrounding these technologies offer invaluable assistance and insights.

As e-commerce continues to evolve, the role of immersive technologies like 3D visualization will only become more prominent. By strategically integrating Three.js and modern JavaScript frameworks into their Shopify stores, businesses can future-proof their online presence, offering engaging and memorable experiences that capture the attention of today's discerning consumers and set them apart from the competition in the years to come.

Works cited

1. How to embed a 3D model viewer in Shopify in 4 easy steps | Visao, accessed April 4, 2025, https://visao.app/3d-model-viewer-shopify/
2. How to Add JavaScript to Shopify: A Comprehensive Guide for Effortless Integration, accessed April 4, 2025, https://www.hulkapps.com/blogs/shopify-hub/how-to-add-javascript-to-shopify-a-comprehensive-guide-for-effortless-integration
3. How to activate 3D Model view automatically in Dawn theme? - Shopify Community, accessed April 4, 2025, https://community.shopify.com/c/shopify-design/how-to-activate-3d-model-view-automatically-in-dawn-theme/m-p/1941883/highlight/true
4. Three JS + Shopify? : r/threejs - Reddit, accessed April 4, 2025, https://www.reddit.com/r/threejs/comments/16v30go/three_js_shopify/
5. 8 Must use JavaScript 3D library 2025 - ThemeSelection, accessed April 4, 2025, https://themeselection.com/javascript-3d-library/
6. 22 Best JavaScript Frameworks to Consider in 2024 (Updated) - TechAhead, accessed April 4, 2025, https://www.techaheadcorp.com/blog/top-22-javascript-frameworks-2024/
7. How To Start With Shopify Storefront API, accessed April 4, 2025, https://scaleshopify.com/2025/01/20/how-to-start-with-shopify-storefront-api/
8. Shopify Website Customization 101: The 2025 Guide for Sellers - FoxEcom, accessed April 4, 2025, https://foxecom.com/blogs/all/shopify-website-customization
9. Custom shopping experiences - Shopify Support, accessed April 4, 2025, https://help.shopify.com/en/manual/custom-storefronts
10. How to Add Custom Javascript to Shopify Store | Full Tutorial 2025 - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=CsxlWpDdw2M
11. Simple Steps to Add JS to Shopify Modules - Brainspate, accessed April 4, 2025, https://brainspate.com/blog/add-js-to-shopify-module/

12. How to develop Shopify themes | Part 10 | Integrating JavaScript and CSS in Shopify Themes | by Muhammad Usman Umar | Medium, accessed April 4, 2025, https://medium.com/@gb.usmanumar/how-to-develop-shopify-themes-part-10-integrating-javascript-and-css-in-shopify-themes-696a05720dd3
13. Where to add custom javascript in shopify 2.0 theme, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/where-to-add-custom-javascript-in-shopify-2-0-theme/m-p/2279926
14. How to Add Custom Javascript to Shopify Theme - eCommerce Thesis, accessed April 4, 2025, https://www.ecommercethesis.com/how-to-add-custom-javascript-to-shopify-theme/
15. How to build an e-commerce storefront with Next.js and Shopify - Bejamas, accessed April 4, 2025, https://bejamas.com/hub/guides/how-to-build-an-e-commerce-storefront-with-next-js-and-shopify
16. BL Custom HTML CSS JS Liquid - Shopify App Store, accessed April 4, 2025, https://apps.shopify.com/bl-custom-html-css-js-liquid
17. Vue.js Tutorial — A Guide on Prototyping Web Apps - Shopify, accessed April 4, 2025, https://www.shopify.com/partners/blog/vuejs-tutorial
18. Top 20 JavaScript Frameworks Comparison in 2025 - Mobilunity, accessed April 4, 2025, https://mobilunity.com/blog/top-20-javascript-frameworks-chosen-by-remote-developers-in-2024/
19. 27 Best JavaScript Frameworks For 2025 - LambdaTest, accessed April 4, 2025, https://www.lambdatest.com/blog/best-javascript-frameworks/
20. How to use Vue.js in Shopify Theme Development (Predictive Search API) - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=p8Te0fdN3hc
21. Master Vue.js 3: The Ultimate Guide for Building Real-World Apps in 2024, accessed April 4, 2025, https://vueschool.io/articles/news/master-vue-js-3-the-ultimate-guide-for-building-real-world-apps-in-2024/
22. Integration with Vue 3 - threejs - Reddit, accessed April 4, 2025, https://www.reddit.com/r/threejs/comments/1i6vdzr/integration_with_vue_3/
23. Awesome 3D experience with VueJS and TresJS: a beginner's guide - This Dot Labs, accessed April 4, 2025, https://www.thisdot.co/blog/awesome-3d-experience-with-vuejs-and-tresjs-a-beginners-guide
24. Shopify Theme Development with Vue.js: Optimise Product & Cart Pages - Skillshare, accessed April 4, 2025, https://www.skillshare.com/en/classes/shopify-theme-development-with-vue-js-optimise-product-and-cart-pages/2031333273
25. Trying to use ThreeJS (R172 and later) inside Shopify using liquid, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/trying-to-use-threejs-r172-and-la

ter-inside-shopify-using-liquid/td-p/2919143
26. How to use ThreeJS with VueJS - Stack Overflow, accessed April 4, 2025, https://stackoverflow.com/questions/55223524/how-to-use-threejs-with-vuejs
27. Trying to use ThreeJS (R172 and later) inside Shopify using liquid, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/trying-to-use-threejs-r172-and-later-inside-shopify-using-liquid/m-p/2919143/highlight/true
28. Going 3D with Trois.js and Vue 3 - VueDose, accessed April 4, 2025, https://vuedose.tips/going-3d-with-trois-js-and-vue-3/
29. troisjs/trois: ThreeJS + VueJS 3 + ViteJS - GitHub, accessed April 4, 2025, https://github.com/troisjs/trois
30. Building an interactive web portfolio with Vue + Three.js — Part Three: Implementing Three.js | by Máximo Fernández | NicaSource | Medium, accessed April 4, 2025, https://medium.com/nicasource/building-an-interactive-web-portfolio-with-vue-three-js-part-three-implementing-three-js-452cb375ef80
31. How can I integrate Vue.js into an existing Shopify theme? - FAUN dev, accessed April 4, 2025, https://faun.dev/c/stories/sdlc_corp/how-can-i-integrate-vuejs-into-an-existing-shopify-theme/
32. Ultimate Guide and Resources on Vuejs and Shopify Theme Development Part 1 - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=CVRikmGlUto
33. Vue.js components in Shopify themes | by Maxim Dyuzhinov - Medium, accessed April 4, 2025, https://js-dev.medium.com/vue-js-components-in-shopify-themes-720eaefb7a2b
34. Hydrogen React - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/api/hydrogen-react
35. Shopify Liquid Development - Using Javascript Frameworks (React, Vue, Svelte, etc), accessed April 4, 2025, https://www.youtube.com/watch?v=qVw1-k7kPEg
36. How We Built Hydrogen: A React Framework for Building Custom Storefronts - Shopify, accessed April 4, 2025, https://shopify.engineering/how-we-built-hydrogen
37. 6 Best Javascript Frameworks for 2025 - Strapi, accessed April 4, 2025, https://strapi.io/blog/best-javascript-frameworks
38. Top JavaScript Frameworks to Learn in 2024 - PixelFreeStudio Blog, accessed April 4, 2025, https://blog.pixelfreestudio.com/top-javascript-frameworks-to-learn-in-2024/
39. The Best JavaScript Framework 2024 - ThemeSelection, accessed April 4, 2025, https://themeselection.com/best-javascript-framework/
40. Vue vs React: What to choose in 2024? - Digitalya, accessed April 4, 2025, https://digitalya.co/blog/vue-vs-react/
41. Vue vs. React: Fundamental 2025 Comparison - IT Craft, accessed April 4, 2025, https://itechcraft.com/blog/react-vs-vue/

42. React vs Vue - YoungWonks, accessed April 4, 2025, https://www.youngwonks.com/blog/react-vs-vue
43. Vue vs React: Which Framework Is Better in 2023 | by sdhglobal | Medium, accessed April 4, 2025, https://medium.com/@sdhglobal/vue-vs-react-which-framework-is-better-in-2023-252acf5bfd92
44. Choosing React or Vue for Your Startup in 2024 - Codica, accessed April 4, 2025, https://www.codica.com/blog/react-vs-vue/
45. Hydrogen: Shopify's headless commerce framework, accessed April 4, 2025, https://hydrogen.shopify.dev/
46. Creating Shopify headless e-commerce in Hydrogen vs Next js | by Hamza 4600 - Medium, accessed April 4, 2025, https://medium.com/@iqrahamzaworks/creating-shopify-headless-e-commerce-in-hydrogen-vs-next-js-6ac5cb166105
47. Building custom storefronts on Shopify with Hydrogen - GitNation, accessed April 4, 2025, https://gitnation.com/contents/building-custom-storefronts-on-shopify-with-hydrogen
48. gitntg/shophydr - Explore custom AI code assistants, accessed April 4, 2025, https://hub.continue.dev/gitntg/shophydr
49. Building Blocks of High Performance Hydrogen-powered Storefronts - Shopify Engineering, accessed April 4, 2025, https://shopify.engineering/high-performance-hydrogen-powered-storefronts
50. Shopify Hydrogen: Build Custom E-Commerce Stores - Naturaily, accessed April 4, 2025, https://naturaily.com/blog/what-is-shopify-hydrogen
51. Updates - Shopify Hydrogen, accessed April 4, 2025, https://hydrogen.shopify.dev/updates
52. Beginners Guide To Shopify Hydrogen (Part 1) - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=RG21VyrF1uw
53. Getting started with Hydrogen and Oxygen - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/storefronts/headless/hydrogen/getting-started
54. Re: Hydrogen with React Three Fiber/DREI libraries? - Shopify Community, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/hydrogen-with-react-three-fiber-drei-libraries/m-p/2739479/highlight/true
55. EduardStroescu/NeuralCouture-Storefront: Headless Shopify Storefront using React-Three-Fiber on the frontend. - GitHub, accessed April 4, 2025, https://github.com/EduardStroescu/NeuralCouture-Storefront
56. Re: Hydrogen with React Three Fiber/DREI libraries? - Shopify Community, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/hydrogen-with-react-three-fiber-drei-libraries/m-p/2925724/highlight/true
57. Build a 3D World in React with ThreeJS and React Three Fiber - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=9ZEjSxDRlik
58. Examples - React Three Fiber, accessed April 4, 2025,

https://r3f.docs.pmnd.rs/getting-started/examples
59. Build a 3D Ecommerce Landing Page with Next.js 14, GSAP, Three.js and Prismic - Full Course 2024 - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=RKQqrNyAC6k
60. Server Rendering & Hydration | TanStack Query React Docs, accessed April 4, 2025, https://tanstack.com/query/latest/docs/framework/react/guides/ssr
61. Understanding Hydration in React applications(SSR) - Saeloun Blog, accessed April 4, 2025, https://blog.saeloun.com/2021/12/16/hydration/
62. Vue.js Hydration Demystified | VueConf US 2024 - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=LdbHmSKp-88
63. Server Side Hydration with Vue.js and SSR - Stack Overflow, accessed April 4, 2025, https://stackoverflow.com/questions/40980009/server-side-hydration-with-vue-js-and-ssr
64. Understand and solve hydration errors in Vue.js - Sapiens, accessed April 4, 2025, https://dach.sapiens.com/understand-and-solve-hydration-errors-in-vue-js/
65. Server-Side Rendering (SSR) - Vue.js, accessed April 4, 2025, https://vuejs.org/guide/scaling-up/ssr.html
66. Client Side Hydration - Vue SSR Guide, accessed April 4, 2025, https://v2.ssr.vuejs.org/guide/hydration.html
67. Everything you need to know about SSR, Hydration and React 18 - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=cVnsxC2g8aE
68. SSR Deep Dive for React Developers, accessed April 4, 2025, https://www.developerway.com/posts/ssr-deep-dive-for-react-developers
69. What is hydration in react? : r/reactjs - Reddit, accessed April 4, 2025, https://www.reddit.com/r/reactjs/comments/18fky3q/what_is_hydration_in_react/
70. Selective Hydration - Patterns.dev, accessed April 4, 2025, https://www.patterns.dev/react/react-selective-hydration/
71. hydrateRoot – React, accessed April 4, 2025, https://react.dev/reference/react-dom/client/hydrateRoot
72. Text content does not match server-rendered HTML | Next.js, accessed April 4, 2025, https://nextjs.org/docs/messages/react-hydration-error
73. How To Fix Hydration Failed Because The Initial UI Does Not Match - DhiWise, accessed April 4, 2025, https://www.dhiwise.com/post/how-to-fix-hydration-failed-because-the-initial-u
74. Solve React hydration errors in Remix/Next apps - Jacob Paris, accessed April 4, 2025, https://www.jacobparis.com/content/remix-hydration-errors
75. Hydration failed because the initial UI does not match what was rendered on the server html, accessed April 4, 2025, https://community.shopify.com/c/shopify-apps/hydration-failed-because-the-initial-ui-does-not-match-what-was/m-p/2652520
76. Hydration Error with Remix and i18next: "Hydration failed because the initial UI does not match what was rendered on the server" - Stack Overflow, accessed April 4, 2025, https://stackoverflow.com/questions/78825007/hydration-error-with-remix-and-i

18next-hydration-failed-because-the-initial-ui
77. About metafields and metaobjects - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/apps/build/custom-data
78. How to work with Metafields when building Shopify themes, accessed April 4, 2025, https://www.shopify.com/uk/partners/blog/metafields
79. Shopify Help Center | Metafields, accessed April 4, 2025, https://help.shopify.com/en/manual/custom-data/metafields
80. How To Create and Use Shopify Metafields in Next.js Commerce, accessed April 4, 2025, https://www.dalicommerce.com/docs/nextjs/create-and-use-shopify-metafields-nextjs
81. Retrieve metafields with the Storefront API - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/storefronts/headless/building-with-the-storefront-api/products-collections/metafields
82. Re: Switch from JS Buy SDK to Storefront API Client Help. - Shopify Community, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/switch-from-js-buy-sdk-to-storefront-api-client-help/m-p/2983457/highlight/true
83. Storefront API - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/api/checkout-ui-extensions/2024-10/apis/storefront-api
84. Issue #168 · Shopify/js-buy-sdk - Product Metafields - GitHub, accessed April 4, 2025, https://github.com/Shopify/js-buy-sdk/issues/168
85. product - Storefront API - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/api/storefront/latest/queries/product
86. Anyway of using Shopify metafield in js.liquid file? - Stack Overflow, accessed April 4, 2025, https://stackoverflow.com/questions/52292287/anyway-of-using-shopify-metafield-in-js-liquid-file
87. How to access a Shopify metafield using JavaScript in a .liquid file? : r/shopifyDev - Reddit, accessed April 4, 2025, https://www.reddit.com/r/shopifyDev/comments/14ik137/how_to_access_a_shopify_metafield_using/
88. Shopify Metafields - How to use Custom Fields - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=cnMwbpJzlQM
89. How to use shopify metafields? · Issue #756 · vercel/commerce - GitHub, accessed April 4, 2025, https://github.com/vercel/commerce/issues/756
90. parseMetafield - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/api/hydrogen/2023-07/utilities/parsemetafield
91. parseMetafield - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/api/hydrogen/2024-10/utilities/parsemetafield
92. Update cart metafields - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/storefronts/headless/hydrogen/cart/metafields
93. three.js examples, accessed April 4, 2025, https://threejs.org/examples/
94. Example store starter template built with Contentful and Hydrogen framework from Shopify - GitHub, accessed April 4, 2025,

https://github.com/contentful/starter-hydrogen-store
95. Re: Trying to use ThreeJS (R172 and later) inside Shopify using liquid, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/trying-to-use-threejs-r172-and-later-inside-shopify-using-liquid/m-p/2919152
96. Re: Trying to use ThreeJS (R172 and later) inside Shopify using liquid, accessed April 4, 2025, https://community.shopify.com/c/technical-q-a/trying-to-use-threejs-r172-and-later-inside-shopify-using-liquid/m-p/2920411
97. Shopify And GitHub Integration - DEV Community, accessed April 4, 2025, https://dev.to/quratulaiinn/shopify-and-github-integration-20a1
98. Liquid objects: metafield - Shopify.dev, accessed April 4, 2025, https://shopify.dev/docs/api/liquid/objects/metafield
99. What Makes Splash Pages Successful? Tips and Examples for Your Website - Shopify, accessed April 4, 2025, https://www.shopify.com/blog/splash-page
100. Best Practices in Splash page Development - Webbook Studio, accessed April 4, 2025, https://webbookstudio.com/articles/best-practices-in-splash-page-development/
101. Shopify Custom Landing Page: Creating Effective and Engaging Landing P, accessed April 4, 2025, https://www.accentuate.io/blogs/wiki/shopify-custom-landing-page-creating-effective-and-engaging-landing-pages
102. Creating high-converting Shopify landing pages (complete guide) | Instant, accessed April 4, 2025, https://instant.so/blog/shopify-landing-pages
103. How to Optimize 3D Models for the Web - Uni Agency, accessed April 4, 2025, https://www.uni.agency/post/3d-web-optimization
104. How to Create High-Performance Shopify Landing Pages? - Cortex Media Marketing, accessed April 4, 2025, https://www.cortexmarketing.in/blog_details/how-to-create-high-performance-shopify-landing-pages
105. How to Optimize WebGL for High-Performance 3D Graphics, accessed April 4, 2025, https://blog.pixelfreestudio.com/how-to-optimize-webgl-for-high-performance-3d-graphics/
106. Building Efficient Three.js Scenes: Optimize Performance While Maintaining Quality, accessed April 4, 2025, https://tympanus.net/codrops/2025/02/11/building-efficient-three-js-scenes-optimize-performance-while-maintaining-quality/
107. Optimizing Three.js Performance for Smooth Rendering, accessed April 4, 2025, https://www.threejsdevelopers.com/blogs/optimizing-three-js-performance-for-smooth-rendering/
108. Better Performance With LOD (Level Of Detail) In Three.js - YouTube, accessed April 4, 2025, https://www.youtube.com/watch?v=IsRBxh4Jb18

109. How can I optimise my THREE.JS rendering? - Questions, accessed April 4, 2025, https://discourse.threejs.org/t/how-can-i-optimise-my-three-js-rendering/42251
110. How to improve performance on my site? - three.js forum, accessed April 4, 2025, https://discourse.threejs.org/t/how-to-improve-performance-on-my-site/52039
111. Improving your online store performance - Shopify Support, accessed April 4, 2025, https://help.shopify.com/en/manual/online-store/web-performance/improving-web-performance
112. Performance best practices for Shopify themes, accessed April 4, 2025, https://shopify.dev/docs/storefronts/themes/best-practices/performance
113. How to Refactor a Shopify Site for Javascript Performance, accessed April 4, 2025, https://www.shopify.com/partners/blog/javascript-performance