VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSI4005

Augmented Reality and Virtual Reality

Winter Semester 2024 -25

Virtual Reality for Store Navigation and Product Display

Faculty: Somasundaram S K

Team 5

21MID0012 Dhruthi G H

21MID0154 Kausalya J

21MID0161 Gowdhami J

21MID0169 Kanishka S J

21MID0181 Samay Raj

Table of contents:

3 Abstract

This project proposes a new virtual reality (VR) shopping experience that uses Three.js and WebXR to recreate an immersive, interactive retail store. The technology attempts to recreate real-world store layouts—complete with aisles, shelves, and products—in a 3D virtual realm, providing users with natural movement and engagement via mouse, keyboard, and VR headset inputs. A notable innovation is its integration with a QR marker-based navigation system, which allows customers to scan aisle-specific QR codes and get extensive product lists directly from their computers or smartphones. Features like search-based teleportation, smooth product interaction, and modular scalability create a seamless and engaging user experience that connects physical and digital retail environments. The system is hosted using Express.js for local development and Netlify for public-facing QR-linked pages, which are tuned for high frame rate and low latency.

4 Literature Survey

4.1. [1] *Shopping in virtual reality: A literature review and future agenda*, Nannan Xi, Juho Hamari: This paper presents a comprehensive literature review on the use of Virtual Reality (VR) in the shopping and retail sector. The authors analyze 72 research papers to map the current state of VR in consumer experiences. It explores various elements such as research methods, devices, technologies, simulated environments, and the psychological impact of VR-based shopping. Using the S-O-R (Stimulus-Organism-Response) framework, the study categorizes the causes and effects of virtual shopping on user behavior. It highlights how immersive environments can influence purchasing decisions and customer satisfaction. The paper also proposes 16 future research directions focusing on methodology, interaction design, and VR content. A key takeaway is the potential of VR to reshape the retail industry through realistic, engaging virtual stores. Limitations such as language scope and database constraints are acknowledged. This research lays a strong foundation for further interdisciplinary work in business and technology.

4.2. [2] *Virtual Reality Shopping-Insights: A data-driven framework to assist the design and development of Virtual Reality shopping environments*, Rubén Grande et al.: This paper introduces Virtual Reality Shopping Insights (VRSI), a comprehensive framework that streamlines the development of VR shopping environments. Built on Unity, VRSI provides developers and researchers with tools to easily integrate VR features and collect user activity data. One of its key strengths is the ability to monitor user behavior through non-invasive tracking, making it useful for marketing analysts aiming to optimize store layouts and user experiences. The framework emphasizes a data-driven approach, enabling evidence-based improvements in V-commerce platforms. The paper also presents a real-world VR shopping application developed using VRSI. Looking ahead, the authors aim to enhance accessibility for users with physical challenges and improve user behavior analysis across multiple sessions. They also plan to integrate social features, like shared shopping experiences and review systems. Overall, VRSI supports both technological and analytical advancements in the evolving VR shopping space.

4.3 [3] Unveiling virtual interactive marketplaces: Shopping motivations in the Metaverse through the lens of uses and gratifications theory, Rajasshrie Pillai et al. (2024) explore the underlying motivations that drive Generation Z to engage in shopping within the Metaverse, using the *Uses and Gratifications Theory (UGT)* as the conceptual foundation. The study proposes a hybrid model that integrates traditional motivational factors such as entertainment, escapism, and social interaction with Metaverse-specific attributes like immersion, personalization, and virtual identity. Employing a sample of 1220 Gen Z consumers and analyzing data through Partial Least Squares Structural Equation Modeling (PLS-SEM) alongside Necessary Condition Analysis (NCA), the research reveals that these gratifications significantly influence shopping intentions, while perceived risk negatively impacts them. Notably, the use of NCA allows the authors to identify not just significant predictors but essential conditions that must be in place for metaverse shopping to occur. The study also finds that personal innovativeness moderates these relationships, suggesting that tech-savvy individuals are more likely to embrace Metaverse shopping. This work provides a dual contribution by extending the UGT framework into immersive technology contexts and offering strategic implications for businesses seeking to attract Gen Z users through enhanced virtual experiences and reduced perceived risks.

4.4 [4] Examining the Moderating Effects of Shopping Orientation, Product Knowledge and Involvement on the Effectiveness of Virtual Reality (VR) Retail Environment Huang et al. (2023) investigate how individual consumer characteristics influence the effectiveness of Virtual Reality (VR) in retail, focusing on shopping orientation, product knowledge, and product involvement. Grounded in experimental research using a between-subjects design across three studies, the authors compare VR-based product presentations with static images to understand their impact on consumer mental imagery and purchase intention. The findings indicate that VR does not consistently outperform traditional formats; instead, its effectiveness is moderated by specific consumer traits. For instance, high product knowledge diminishes the advantage of VR in facilitating mental imagery, but this effect is mitigated when consumers are shopping with an experiential rather than task-focused orientation. Additionally, high product involvement also helps counteract the negative influence of product knowledge on VR effectiveness. This study makes significant contributions by emphasizing that VR is not a universally superior tool in retail settings. Rather, its success depends on aligning the technology with consumer orientation and product context. The work offers practical implications for marketers, suggesting the need for strategic implementation of VR tailored to product types and target consumer profiles.

4.5 [5] Does Imagination Compensate for the Need for Touch in 360-Virtual Shopping? Ruusunen et al. (2023) explore how consumers' sensory needs and cognitive abilities interact in 360-virtual shopping environments, particularly focusing on whether imagination can compensate for the absence of physical touch. Drawing on the theory of embodied cognition, the study investigates the roles of telepresence, the instrumental and autotelic dimensions of the need for touch, and consumer imagination. Using a sample of 900 participants, the authors find that while telepresence enhances consumer attitudes toward virtual shopping, this positive effect is significantly diminished for individuals with a high autotelic (hedonic) need for touch. However, imagination is shown to partially compensate for the lack of touch, but only in relation to the instrumental (utilitarian) need for touch. The study identifies key boundary conditions, revealing that imagination cannot offset hedonic sensory desires in virtual environments. The research offers practical implications for marketers designing 360-virtual stores, suggesting that interactive visual features—such as the ability to rotate products—can stimulate haptic imagination and improve the perceived functionality of products. These findings highlight that while imagination has the potential to bridge sensory gaps in virtual shopping, it is more effective for functional than experiential product evaluation.

4.5 Drawbacks from Existing System

Although virtual shopping environments, particularly those leveraging Virtual Reality (VR), 360-degree views, and immersive technologies, offer promising enhancements over traditional e-commerce, several critical drawbacks persist in current implementations.

Inability to Fully Replicate Physical Sensory Experience Virtual shopping lacks tactile interaction, which is essential for evaluating product quality and deriving sensory satisfaction. Imagination can only partially fill this gap, especially for hedonic needs.This leaves many consumers unconvinced and disconnected from the product. Limited Consideration of Individual Differences Most systems ignore key user traits like shopping orientation or product involvement. This reduces engagement, as one-size-fits-all experiences miss personal relevance. Customization is crucial, especially for digital natives like Gen Z.

Restricted Product Categories and Contexts Current systems focus on limited product types under controlled conditions. This undermines the scalability and real-world effectiveness of VR shopping. Diverse categories need different sensory and emotional cues to work. High Cognitive Load and Accessibility Barriers. Immersive systems often demand high digital literacy and tech comfort. Users unfamiliar with VR tools may feel overwhelmed or excluded. Hardware costs also limit access for broader user groups. Underutilization of Imagination and Mental Imagery

Existing platforms don't actively stimulate the user's mental imagery. This weakens emotional engagement and reduces the immersive potential. Especially for hedonic products, imagination alone isn't enough. Neglect of Multisensory and Emotional Design Most systems focus on visuals, overlooking sound, motion, or story. This lack of emotional layering weakens user connection to the experience.Multisensory cues are essential for presence and product appeal.

5 Software and Hardware Requirements

5.1 Software requirement

Three.js The JavaScript library Three.js is designed to produce real-time 3D graphics in web browsers. It provides the core elements for a scene within a VR store by facilitating the creation of scenes setup, loading and positioning 3D product models, lighting, and control interactivity. Additionally, it offers advanced raycasting for user interaction as well as shader programming capabilities for visual enhancement.

WebXR API This is the API connecting web apps with VR/AR gear beside or blended with the use of Three.js to immerse users of VR-compatible headsets into their environment. It will preload apply powerful stereo-rendering, which itself includes VR-specific input types, to experience the shop in a more real and interactive condition when fitted with compatible headsets.

Express.js Express.js is a lightweight and yet flexible web application framework for Node.js mainly used for local DEVs server apps. The VR store application runs with Express as its static asset-serving function, delivering such assets as HTML, CSS, JavaScript, and 3D models in an efficient way. The express smoking duty applies especially when developing the application because then it provides a quick backend where access to the application is "developed" and "deployed" quickly for immediate testing.

Node.js Node.js is essentially a runtime environment that serves for running the Express.js server. It manages backend setups and processes through tools that package and dependency management using npm (Node Package Manager). Using Node.js guarantees that all modules would run and integrate properly with the development workflow of the VR application.

Netlify Netlify formulates the implementation of QR marker-based product information systems. It is a contemporary cloud platform that serves for hosting static front-end applications. Netlify has the following benefits- automatic HTTPS, global CDN delivery, easy connectivity with version control to provide continuous deployment. This makes it a good option for hosting marker-based modules of a VR project.

Raycasting with 3js it is a particular method in Three.js which detects user interaction within a 3-dimensional space. For example, it is able to tell which object within the 3D scene has been pointed to, based on the interaction brought on by user input such as a mouse click. This enables an ever-dynamic experience with products and QR markers, as the system can recognize the user selection and respond accordingly.

5.2 Hardware requirement:

Computer or Laptop Desktop The modern computer or laptop would serve dual purposes in running the application as it would have compatible specifications including Intel i5 or AMD Ryzen 5 CPU (and higher), 8 GB RAM at least (16 GB recommended), and a dedicated graphics processing unit such as NVIDIA GTX 1650 or better. A more modern web browser like Chrome, FireFox, and Edge can employ WebXR in running the VR interface well.

A Mobile Device with Camera This mobile device should ideally have a functioning camera within it in order to scan the QR markers placed within the virtual aisles. Either with QR code scanning feature in-built in browser or with third-party applications installed within, a mobile device should have such functionality to make the marker-based module of the store much more interactive.

Graphics Processing Unit (GPU) Although the application can technically run on integrated graphics, the recommendation is to include a dedicated GPU. This builds towards rendering 3D scenes at a much faster pace, accommodating multiple 3D models, and providing smooth navigation and transitions. This is significant in keeping a virtual reality experience immersive.

Internet Connection A stable internet connection is needed to access content deployed on platforms like Netlify. The connection also allows the downloading of external assets, such as 3D models, textures, and JavaScript libraries, if not stored locally. Internet access allows smooth functionality and loading of assets throughout the application.

6 Proposed System

6.1 Objective of the Project
The main purpose of the project is to build a Virtual Reality (VR) store environment utilizing Three.js and WebXR wherein a user senses a virtually reachable 3D shopping space, interacts with items, and obtains in-depth details. The VR simulated store environment replicates actual shopping by modeling store data such as aisles, shelves, and products and adding interactive functionalities through mouse, keyboard, or VR headsets. The customer can select and compare products in depth, inspect individual products, and even teleport to the actual location of the product. For quicker discovery of products, the project is also linked with a QR system-based system where QR markers labeled at aisle locations enable users to scan rapidly and view a detailed list of products in an aisle using their phone or browser.

6.2 Novelty of the Project
The project presents a new combination of Web-based VR (Three.js + WebXR) and QR marker recognition system to provide in-store experience as well as easy contextual product discovery. Its innovativeness is: Combining Immersive VR Shopping with Physical Shelves: Rather than using current 2D e-commerce, shoppers can shop in a virtual store, which makes them feel present and know where things are. QR Marker-Based Product Discovery: Scanning in-world or on-screen QR marker reveals connected aisle product details, reducing search time and increasing convenience. Platform Independence: The mobile/desktop browser and VR headset host but do not natively support mobile apps, and reach is wide as a result. Search-Driven Movement: Users can search product names and teleport their precise 3D locations, fusing natural interaction and direct teleportation. Modular and Expansible: The design supports scaling—new products, new markers, and aisles can be introduced with minimal code modifications.

6.3 Features / Advantages of the Project

The virtual shopping platform holds many advanced features to improve the user experience and performance of the system. Virtual stores perfectly simulate the layout inside a real shop, modeled with product aisles, shelf rows, and so on, with realistic electrical lights creating an ambience similar to that of fake stores. With assets coming from Sketchfab, the system creates rich visuals to achieve a finished and photo-real feel in the products and interiors of the store.

Movement is very customizable for the user via keyboard keys (WASD), mouse functions through OrbitControls, or VR headset controls with WebX. These various forms of input allow the system to be universal for all users regardless of the device used. Movement is quite smooth as the user transitions between store sections almost cinematically transporting the user from one side of the store to another.

A core highlight is product interaction. A user may click on any product view an information panel displaying the product's name and description. This interaction is intuitive and models the experience of picking up a product in a physical store. In addition, a search feature allows users to type in a product name and then the camera will animate zooming directly into its position to reach the product in question.

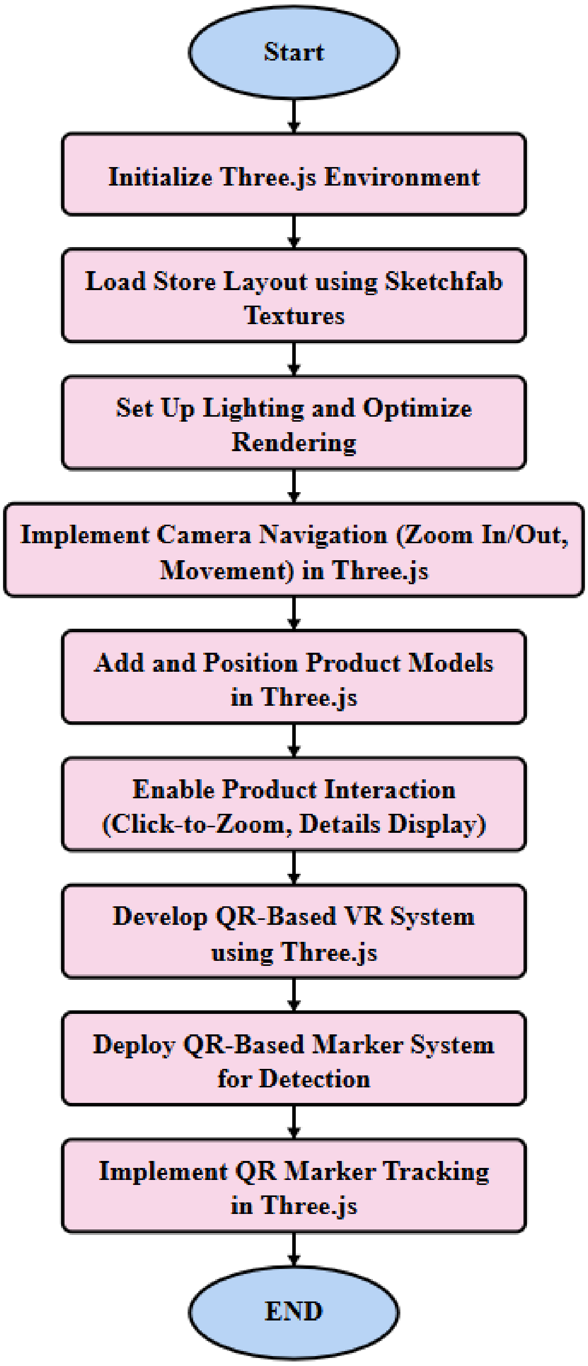
The system also incorporates a QR marker-integration where each aisle has a unique QR code assigned to it. When a QR code is scanned by the user, it will redirect them to a webpage (hosted on NETLIFY) containing a list of all items in that aisle. This gives users a good advantage because manual search within the shop becomes unnecessary; finding what they are looking for is just faster and far cleaner.

In the meantime, the application operates in development mode locally on Express.js, while NETLIFY deploys the public QR-related pages. The focus here is on performance optimization-Three.js rendering is optimized for high frame rates with the least possible latency, securing a super smooth VR experience. The modular nature of the system allows any developer to quickly add new products, modify the store layouts, or even implement new functions without changing the core logic of the system. This guarantees maintainability and scalability.

The procedure initiates with a user who accesses the platform via a web browser or a VR headset. Upon entering the virtual environment, the Three.js engine invokes scene assets that include shelves, aisles, and products. The assets are navigable using the keyboard, mouse, and VR input devices. Users may interact with products in exploration by clicking or selecting them to obtain real-time details about the product. Product searching may be executed by the user, upon which active animation moves the camera to the product location. Scanning of QR markers along the aisles will open external product detail pages for smoother navigation and hybrid integrations. At the backend, product metadata and rendering logics are dealt with through preloaded scripts and Express.js servers, while static product pages are served from Netlify. Optimization mechanisms secure rendering smoothly from multiple areas along with less latency.

6.4 Workflow diagram

Workflow begins with user input to the platform through web browser or through a VR headset. Upon user input to virtual space, Three.js engine loads scene assets like shelves, aisles, and products. Users may then move on to interact using keyboard, mouse, or through VR input. As users navigate, they are able to interact with products using click or tap and product information is updated in real time. They are able to scan for specific products, and the camera will automatically pan to exactly where the product is. QR tags placed along aisles may be scanned in order to open external product detail pages to speed up navigation and hybrid convergence. Backend product metadata and rendering logic are controlled using preloaded scripts and Express.js servers. Optimization mechanisms guarantee smooth rendering and low latency overall.



6.5 Description of Modules

Module 1-VR Store Environment and Rendering Module This module is responsible for creating and rendering the entire virtual store environment. The 3D scene is created with Three.js initialization, followed by the camera- and renderer-adaptation, and WebXR support for VR interaction. Different 3D assets, such as walls, shelves, and floor textures, are loaded via asset loaders, and ambient and directional lighting set up to give a realistic touch to the shopping experience. Visual fidelity has been maintained by sourcing assets from platforms such as Sketchfab. A rendering loop is optimized for keeping a higher framerate in an immersive VR experience.

Module 2-Product Interaction and Navigation Module This module covers the navigation and interaction aspects of the user. It supports various input types-from basic keyboard navigation with the WASD keys and mouse navigation through OrbitControls, to movement using a VR headset via WebXR. All products are constructed in a 3D grid system and when clicked, an item would highlight and present its information. Zoom in allows the user to physically get up close with the objects. This gives the extra navigation based on search; when a product name is entered, the camera will smoothly animate to the product's 3D coordinates.

Module 3: Marker-based Interactivity (QR) This module deals with the context-specific navigating using QR codes. Simply every aisle would have a unique QR marker such that it can be scanned interactively or outside a linked device. Upon scanning, the system immediately redirects to a website hosted up on Netlify, giving information about products in that aisle, their locations, and some short descriptions. Therefore, products become easier to discover, providing a hybrid interface billowing the virtual experience with physical interaction.

Module 4: Server & Deployment Module This module goes on to take care of all hosting, data handling and deployment logistics. When this was under development, a local server could be created using Express.js and the public-facing QR pages are put online using Netlify. The product data and the 3D positions are kept in a highly structured positions.js file, allowing for a quick lookup and updates. The styling takes care of a full-screen and responsive style.css which caters to both desktop modes and VR display. To ensure responsiveness on all devices, the optimization will also involve implementing preloaders, lean imports of modules, as well as resource management strategies.

7 Implementation

Code:

```
//imported libraries

import * as THREE from "three";

import { GLTFLoader } from "three/examples/jsm/loaders/GLTFLoader.js";

import { VRButton } from "three/examples/jsm/webxr/VRButton.js";

import { OrbitControls } from "three/examples/jsm/controls/OrbitControls.js";
```

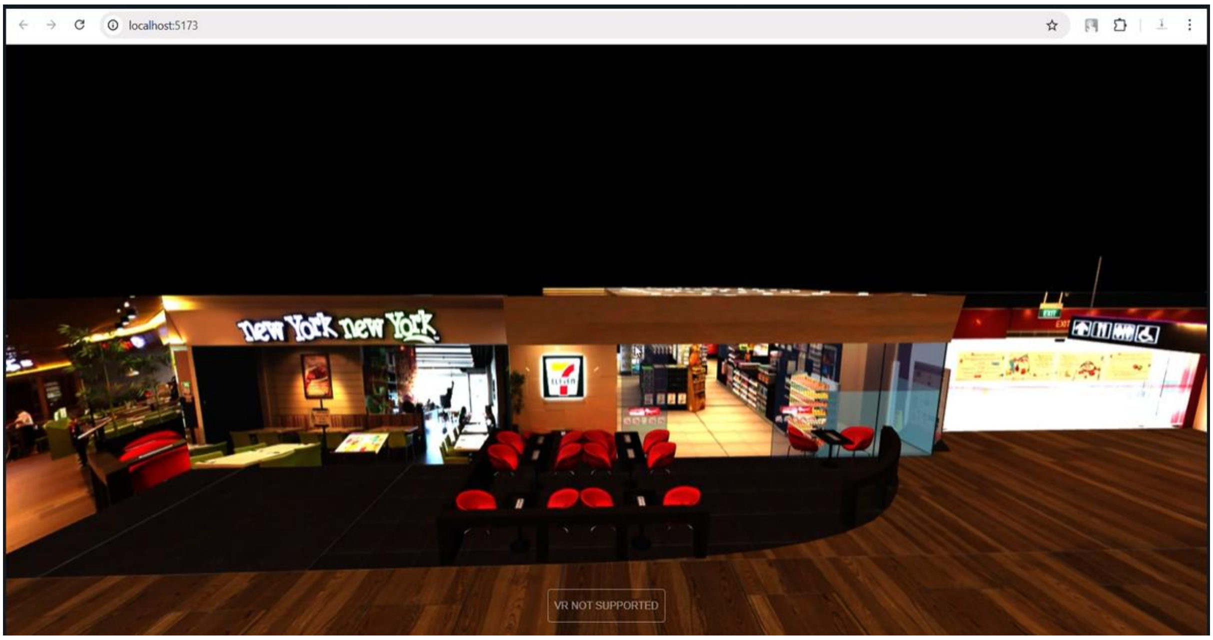
```
import { ProductInteraction } from "./product-interaction.js";
// Product data with prices
    this.productData = {
      Chocolates: { price: 29.99, position: new THREE.Vector3(1, 1.6, -2) },
      "Self-Help Books": {
        price: 19.99,
        position: new THREE.Vector3(-1, 1.6, -3),
      },
      "Phone Top-Up": { price: 39.99, position: new THREE.Vector3(2, 1.6, -4) },
      "Dairy and Cold Beverages": {
        price: 49.99,
        position: new THREE.Vector3(-2, 1.6, 2.5),
      },
      Chips: { price: 24.99, position: new THREE.Vector3(-3, 1.6, 1) },
      "Ice cream": { price: 34.99, position: new THREE.Vector3(-5.5, 1.6, -3) },
      Coffee: { price: 2.5, position: new THREE.Vector3(4, 1.6, -2) },
      "Soft Drinks": {
        price: 59.99,
        position: new THREE.Vector3(-3, 1.6, -3.6),
      },
    };
WASD Keys navigation
window.addEventListener("keydown", (event) => {
  if (event.key === "w") moveCamera("forward");
  if (event.key === "s") moveCamera("backward");
  if (event.key === "a") moveCamera("left");
  if (event.key === "d") moveCamera("right");
```

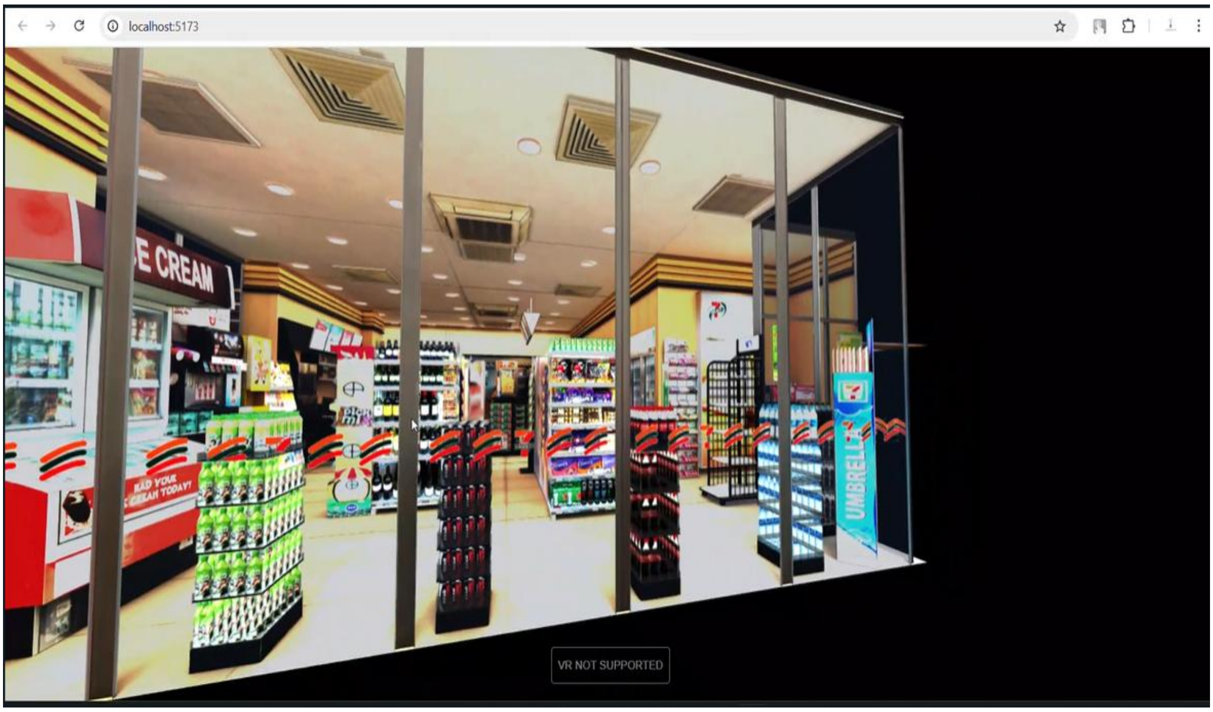
```
});
// Scene Setup
const scene = new THREE.Scene();

// Camera Setup
const camera = new THREE.PerspectiveCamera(
  75,
  window.innerWidth / window.innerHeight,
  0.1,
  1000
);
camera.position.set(10, 1.6, 0.4);
scene.add(camera);

// Renderer
const renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.xr.enabled = true;
document.body.appendChild(renderer.domElement);
// Lighting
const ambientLight = new THREE.AmbientLight(0xffffff, 1.2);
scene.add(ambientLight);
const directionalLight = new THREE.DirectionalLight(0xffffff, 1);
directionalLight.position.set(2, 2, 2);
scene.add(directionalLight);

// Load VR Store Model
```

```
const loader = new GLTFLoader();
loader.load("./assets/scene.gltf", function (gltf) {
  const model = gltf.scene;
  model.position.set(0, 0, 0);
  model.scale.set(1, 1, 1);

  // Improve Visibility
  model.traverse((child) => {
    if (child.isMesh) {
      child.material.emissive = new THREE.Color(0x444444);
    }
  });

  scene.add(model);
});
```
The above code is just a part of the project.

8 Results

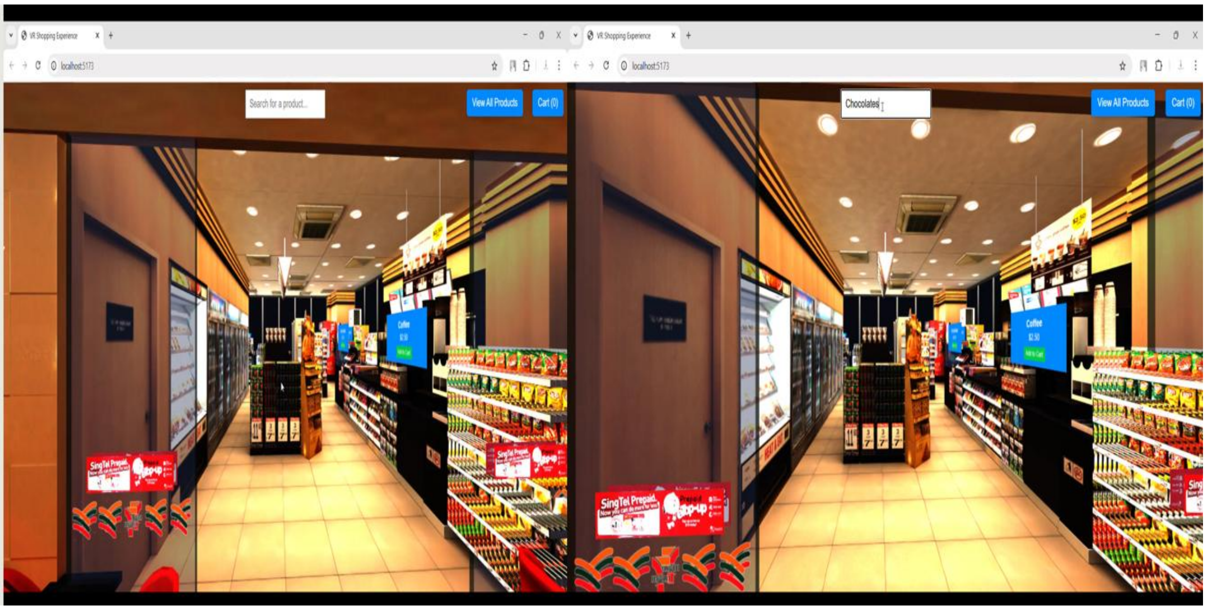
VR Store:



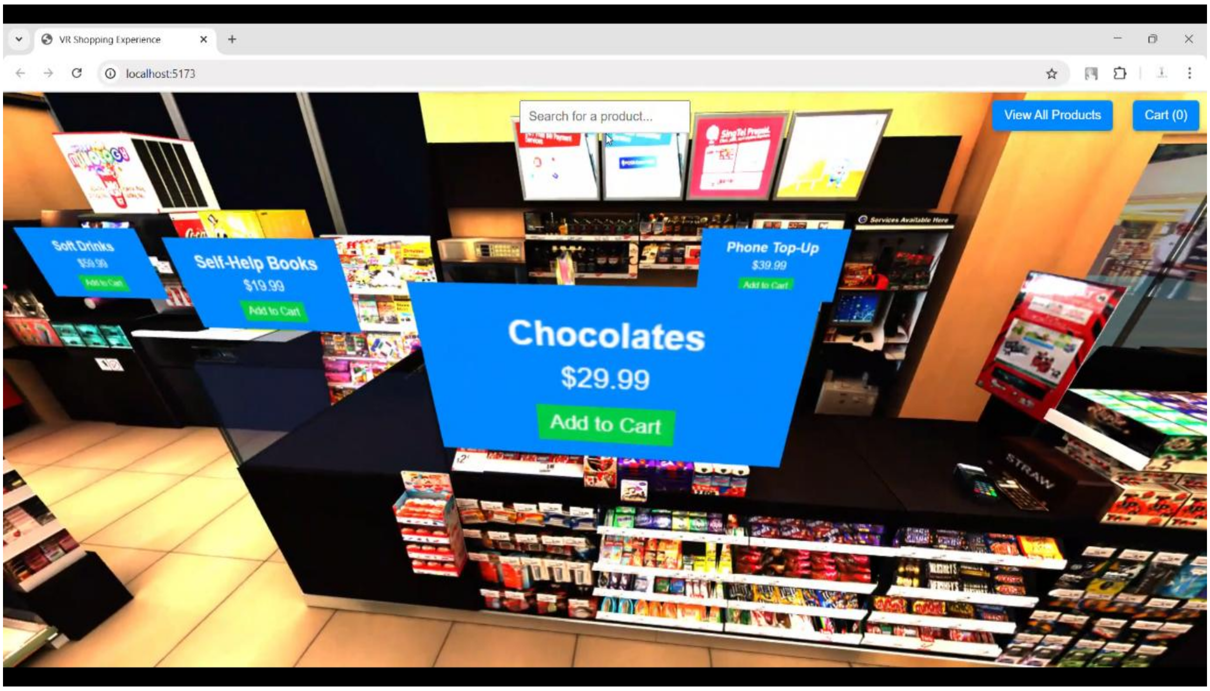


Search Products:

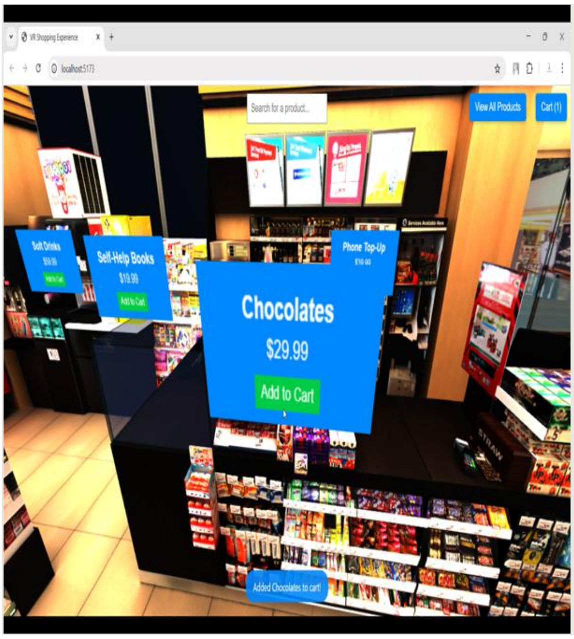
User can search products in the VR store where it navigates to the product.

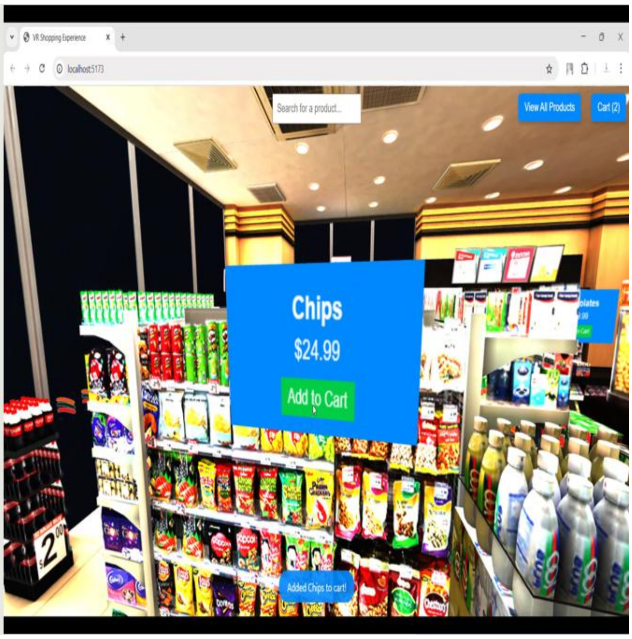


When user searches chocolate, it navigates to the chocolate aisle

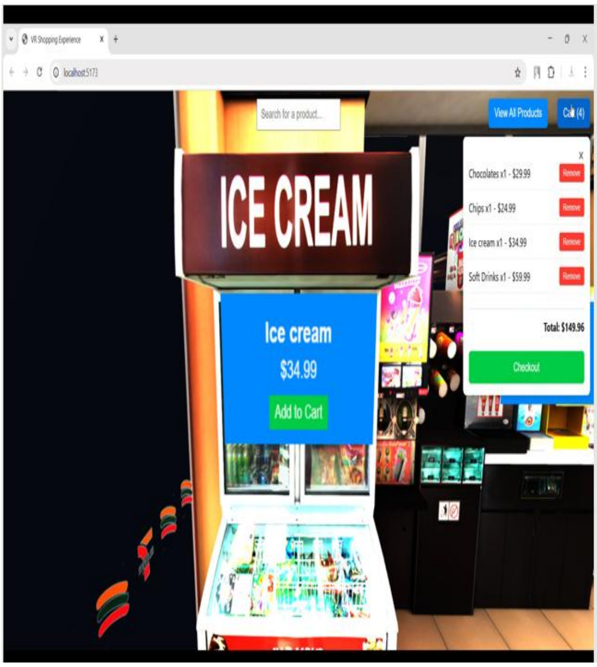


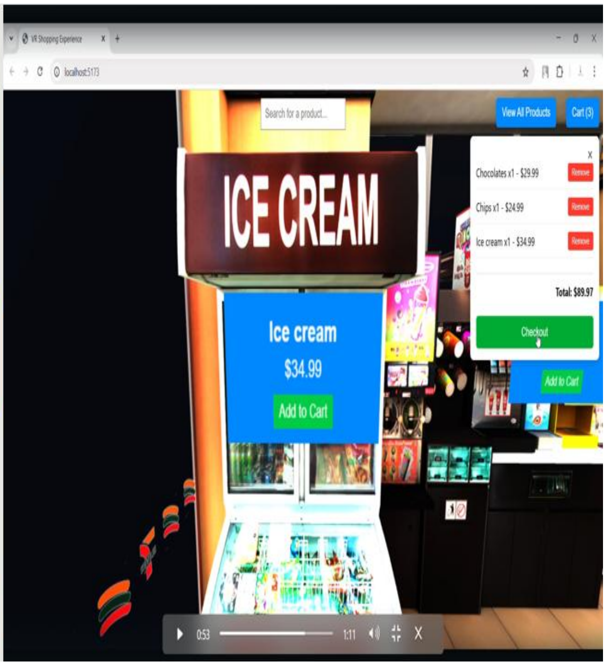
User can add products to the cart.



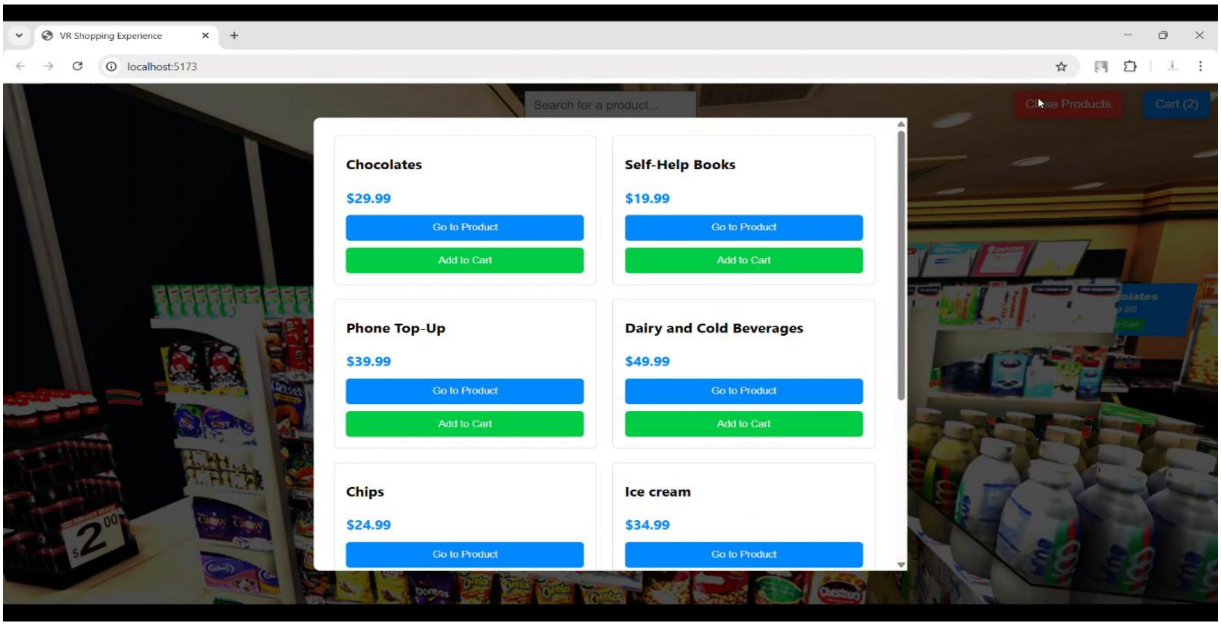


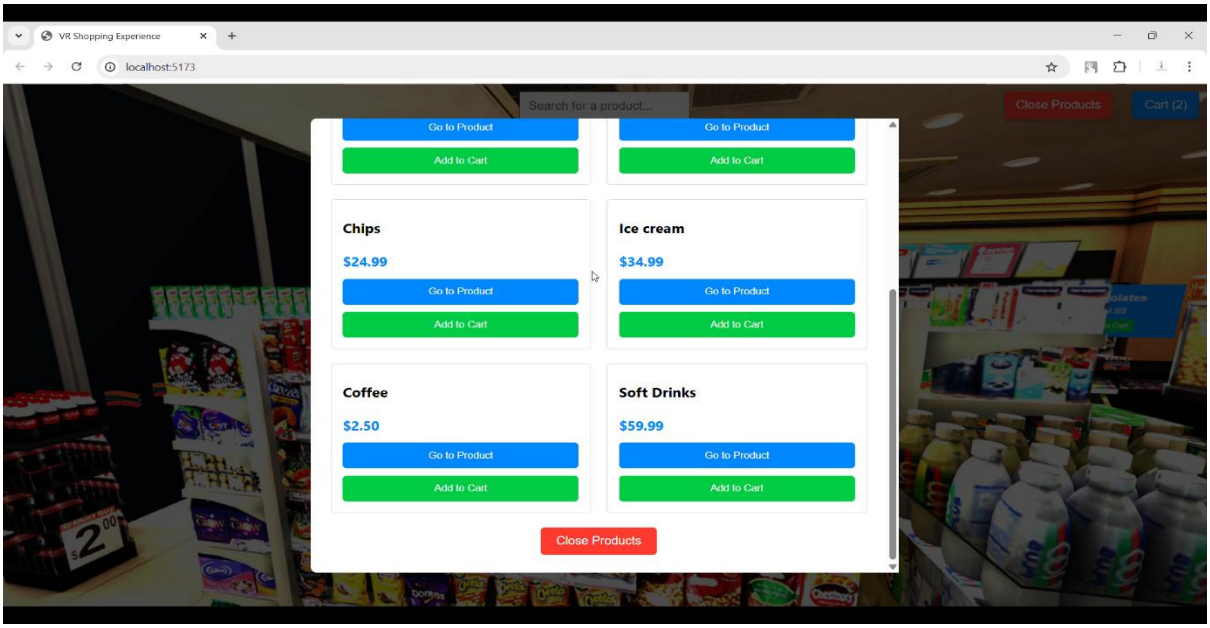
User can view, add, and remove products from cart.



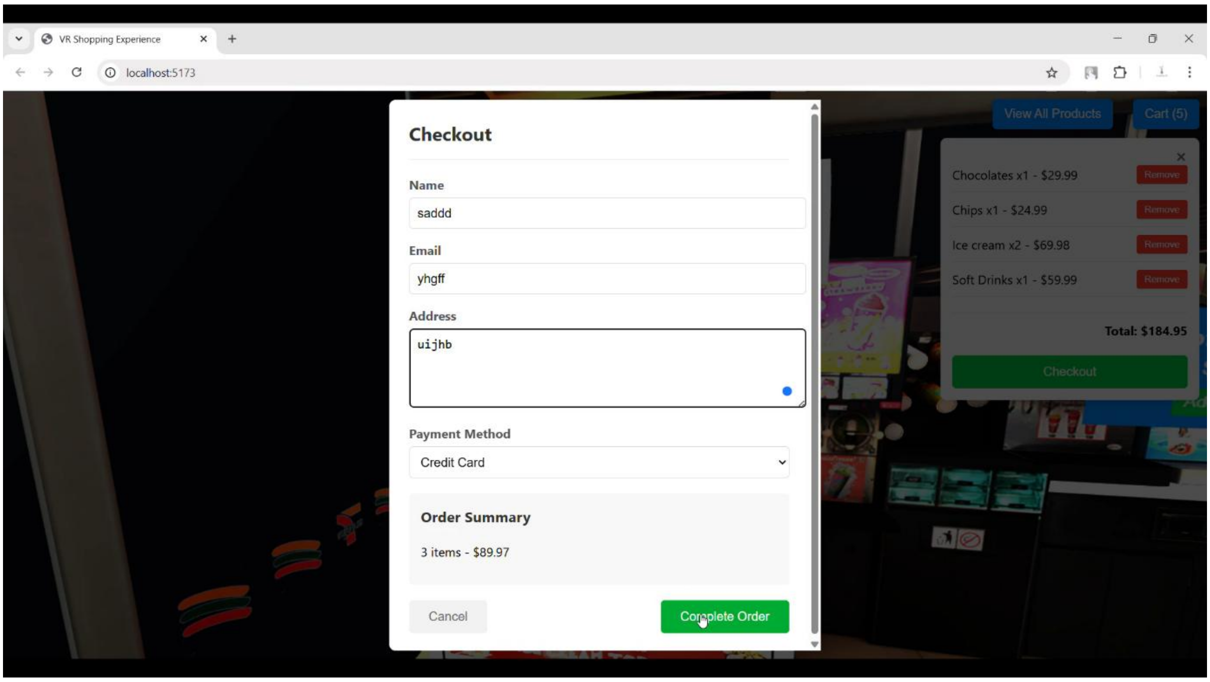


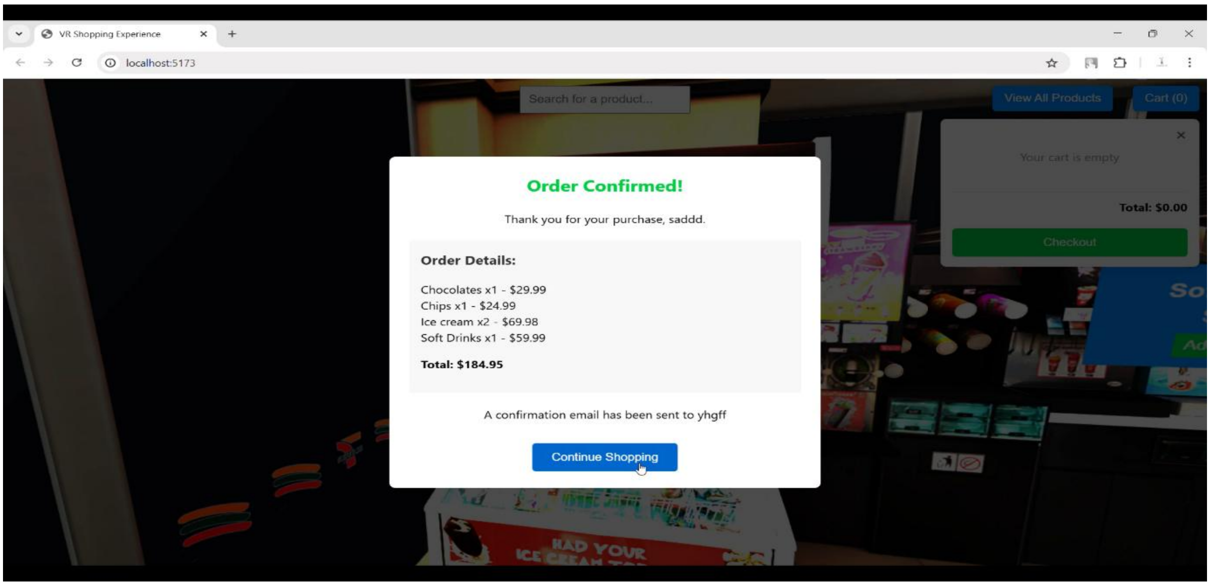
User can view all products in the store:





User can complete their checkout and confirm order by giving necessary details





QR-Based System:

9 Conclusion

This study successfully demonstrates how immersive technologies like Web-based VR, when combined with contextual tools like QR markers, may transform digital retail by mimicking the ease of physical buying. The suggested technology improves the user experience through realistic virtual navigation and product engagement while also reducing search friction, allowing for faster and more informed decisions. Its modular and scalable design assures that it can adapt to changing retail needs while also providing extensive cross-platform interoperability and easy content management.

Future advancements will look into Augmented Reality (AR) extensions that combine in-store experiences with AR overlays on mobile devices, enabling hybrid physical-virtual navigation. Additional capabilities such as voice-based search, real-time inventory integration, multi-user collaboration in VR, and AI-driven product recommendation systems will enhance functionality. The long-term goal is to deploy the system in real-world retail chains as a plug-and-play solution to improve digital transformation in the shopping arena.

10Reference

[1] N. Xi and J. Hamari, "Shopping in virtual reality: A literature review and future agenda," *Journal of Business Research*, vol. 134, pp. 37–58, 2021, doi: 10.1016/j.jbusres.2021.05.016.

[2] R. Grande, J. A. Albusac, D. Vallejo, C. González-Morcillo, S. Sánchez-Sobrino, and J. J. Castro-Schez, "Virtual Reality Shopping-Insights: A data-driven framework to assist the design and development of Virtual Reality shopping environments," *Smart Health*, vol. 34, p. 100473, Dec. 2024, doi: 10.1016/j.smhl.2024.100473.

[3] Unveiling virtual interactive marketplaces: Shopping motivations in the Metaverse through the lens of uses and gratifications theory": S. Kim, Y. Kim and J. Lee, "Unveiling virtual interactive marketplaces: Shopping motivations in the Metaverse through the lens of uses and gratifications theory," *Journal of Business Research*, vol. 168, pp. 114–125, 2023, doi: 10.1016/j.jbusres.2023.06.017.

[4] Y. Huang, L. Mattila, and Y. Liu, "Examining the moderating effects of shopping orientation, product knowledge and involvement on the effectiveness of Virtual Reality (VR) retail environment," *Journal of Business Research*, vol. 160, pp. 113702, 2023, doi: 10.1016/j.jbusres.2023.113702.

[5] N. Ruusunen, H. Hallikainen, and T. Laukkanen, "Does imagination compensate for the need for touch in 360-virtual shopping?" *Information & Management*, vol. 61, no. 1, p. 103848, 2024. [Online]. Available: https://doi.org/10.1016/j.im.2023.103848