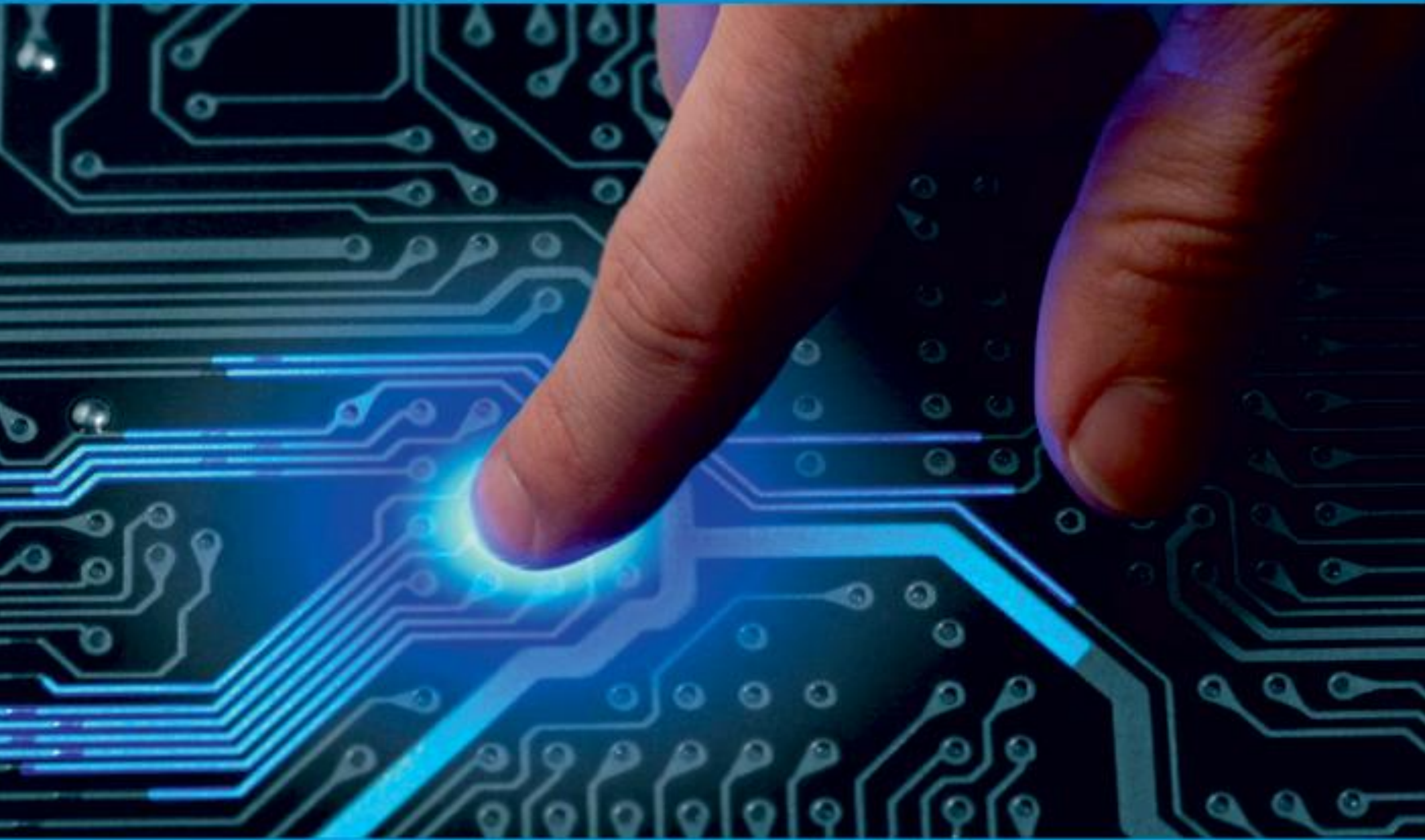
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 5, May 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105037 |

3D Product Configurator Web-Tool

Ayushman Pandey, Shikha Singh, Bramha Hazela, Garima Srivastava

Department of Computer Science and Engineering, Amity School of Engineering and Technology,

Amity University Uttar Pradesh, Lucknow, Campus India

ABSTRACT- A product configurator is a piece of software which is enabling users to alter a product's appearance. The configurator displays every product variation that could be made. Customers can utilise a website's capability to customise a product. A configurator may be used by customers to alter designs, sizes, or colours.
This paper presents a 3D web-based model customization website that enables users to create and customize 3D models in a web browser using WebGL and Three.js technologies. The site offers a wide range of 3D models, and users can choose different materials, textures, and adjust the size and position of the model as per their requirements. The website also features a simple interface for previewing and ordering the customized models, which can be exported in various file formats for use in other applications. This solution provides an accessible alternative to traditional 3D modeling software, allowing users to create and customize 3D models without the need for specialized software or hardware.The 3D product configurator allows customers to personalize a product's color, size, and other features, with the customized model displayed on the screen in real-time. This interactive experience can better engage potential buyers and increase sales. The 3D models can also be used for creating physical objects using various printing services available online. The use of game engines and other 3D rendering technologies enables businesses to create and stream interactive customer experiences to any device, eliminating the need for a physical store or showroom. However, creating a fully dynamic and reliable 3D experience online requires expertise that most businesses lack.

KEYWORDS- 3D Web Viewing, Product Development, e-Commerce, WebGL, computer 3D graphic.

I. INTRODUCTION

The use of web applications has increased, resulting in a decrease in standalone applications in recent years. However, the majority of graphic material is still displayed in 2D. The introduction of WebGL[1] has allowed for the presentation of visually demanding applications on web browsers, which has made many apps more accessible and eliminated the need for installation. The purpose of this report is to evaluate how effectively leading 3D web technologies perform tasks that would have previously been done by standalone applications, and Three JS was selected for this evaluation.

The report covers the basics of presenting 3D visuals, starting with renderers and progressing to sceneries, lighting, textures, and models. The 3D Product Configurator is a useful tool that does not require bulky hardware, making it more accessible than resource-hungry programs such as AutoCAD. Users can access the tool with a device and view models in an interactive 3D space. The tool is also systematic and secure, with a unique ID required for login. The page remains fully secure and does not manipulate user data, and it is free to use, making it accessible to most people.

The Three JS 3D library [2] is one of the most well-known libraries of its kind and is introduced in the report. Some of its capabilities and notable applications are also discussed. The report outlines the project's specifications and its real development, as well as a literature review of the work done in this field. The methodology adopted for solving the problem is described, along with a brief explanation of the graphic processing. The development is explained in technical detail, including the choices made and the challenges overcome. The report concludes by evaluating how successfully the predetermined requirements for the job were met and assessing the effectiveness of Three JS for this type of project.

In conclusion, the use of Three JS for 3D web technology has made visually demanding applications more accessible and eliminated the need for standalone applications[3]. The 3D Product Configurator is a useful tool for interacting with 3D models and is both systematic and secure. The report provides valuable insight into the development and capabilities of Three JS and assesses its effectiveness for this type of project.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105037 |

II. Literature Review

The use of three-dimensional (3D) virtual reality systems has grown to be a well-liked topic for study and development inrecent years, particularly in the field of e-commerce. This literature review will examine various studies that focus on the use of 3D technology in the customization and visualization of products in the online retail industry.

A paper included a study to evaluate the technology acceptance model for 3D virtual reality systems in luxury brand online stores. They discovered that using virtual reality technology enhances the user experience, leading to increased customer satisfaction and brand loyalty [4].

Another researcher developed an interactive web system for integrated 3D customization, which aimed to improve the user experience of online ordering systems. The study found that the system was successful in facilitating efficient and user-friendly customization of products [5].

The authors, showed an intelligent real-time 3D configuration platform for customizing E-commerce products. The platform was designed to provide users with an interactive and intuitive experience in customizing products online [6].

In the research it was proposed a web-based collaborative framework for facilitating decision-making in the 3D design development process. The framework was shown to be a useful tool for enhancing the effectiveness and effective tool for improving the efficiency and quality of the design process [7].

Potenziani in his paper developed the 3D Heritage Online Presenter (3DHOP), a tool for delivering and presenting 3D heritage models on the web. The study found that 3DHOP is a useful tool for making 3D heritage models available to a wider audience [8].

Research examined the readiness of the web for delivering and using 3D models. The study found that while progress notwithstanding the progress that has been achieved in this area, moreimprovement in terms of accessibility and usability [9].

A model was dsigned and developed a 3D digital cadastre visualization prototype. The study found that the use of 3D technology improved the visualization and understanding of cadastre data [10].

Research conducted a survey and evaluation of 3D mass customization toolkits. The study found that the use of these toolkits can improve the efficiency and effectiveness of the customization process [11].

The study which highlights the importance of online product configuration in e-commerce with the use of 3D web viewing technology. According to the study, the use of 3D technology has led to an increase in customer satisfaction, as it allows them to visualize and interact with the products before purchasing. This study provides insight into the benefits of 3D technology in e-commerce and how it can help in improving the customer experience [12].

In the study, the authors provide a comprehensive guide on the conversion process from CAD (Computer-Aided Design) to VR (Virtual Reality). The authors argue that the conversion of CAD models to VR can be a valuable tool for product visualization in e-commerce. They suggest that by using VR, customers can have a more realistic and immersive experience with the product [13].

The research study focuses on the lean startup orientation and its effect on venture success. The study provides empirical evidence on the importance of the lean startup approach in e-commerce and how it can help in reducing the risk of failure. This study highlights the importance of a lean and efficient approach in e-commerce, which can be achieved through the use of technology such as 3D web viewing [14].

The paper provides insight into the development and implementation of a three-dimensional online customization ordering system. According to the writers, utilising this technology can assist to enhance the client experience. as it allows customers to customize products to their specific needs. The study highlights the potential of 3D technology in product customization in e-commerce [15].

Finally, the study which focuses on the use of reality-based 3D modelling, segmentation, and web-based visualization in digital heritage. The authors argue that the use of 3D technology can help in improving the representation of protecting cultural treasures for future generations. The study provides insight into the potential of 3D technology in preserving cultural heritage and its importance in providing an immersive and interactive experience for users [16].

In conclusion, the studies cited in the review demonstrate the potential and benefits of the use of 3D web viewing technology in e-commerce. From product customization and visualization to venture success and cultural heritage preservation, the use of 3D technology has proven to be a valuable tool in enhancing the customer experience and improving the efficiency of e-commerce processes.

III. 3D GRAPHIC BASICS

An insight on how the 3D customization made to the merchandise:

Rendering
The first component needed is a renderer, which is responsible for displaying the graphics on the screen. Rendering is a challenging process that involves converting a scene's mathematical data into a 2D picture. There are two forms of rendering: pre-rendering and real-time rendering. Pre-rendering is typically used to create videos or single images, while real-time rendering is used in applications such as games where the next picture to be produced is unpredictable. Real-time rendering can produce visuals that are considerably better than pre-rendering [17].

Scene
A scene is another important component required for 3D customization. All the information about the items in the world that are intended to be rendered is contained in a scene. The scene can be viewed through a camera, which is used to monitor the environment or the "scene." The renderer displays what is visible through the camera.

Mesh, texture and material
Mesh is one of the most commonly added objects to a scene, and it holds information on the shape of the object. Textures and materials may be applied to meshes to give them the desired appearance. For example, if a wooden table had to be added to a scene, a mesh resembling a table would first be created. Next, a texture— an image of a wooden table's surface—would be applied to it to make it look like it's made of wood. However, it wouldn't reflect light as a hardwood table would. The mesh is given a material to adjust how its surface reacts to light.

Coordinate System
To place the generated object in a 3D environment, a coordinate system is necessary. The most commonly used coordinate system is known as Cartesian Space, which defines horizontal, vertical, and depth positions of an object using the terms x, y, and z, respectively. The location of the object can be expressed as a vector with respect to the origin of the space. Scalars that describe a point's location along a basis vector make up the points in a tuple that define a position in a 3D coordinate system. A 3D space has three basis vectors, which are mutually perpendicular and have a length of exactly 1, making them unit vectors [18]which is represented in **Figure 1**:
Each of these vectors must have a length of exactly 1, making them unit vectors [18].
The coordinate system's point (1,1,1) would be located at the following location:

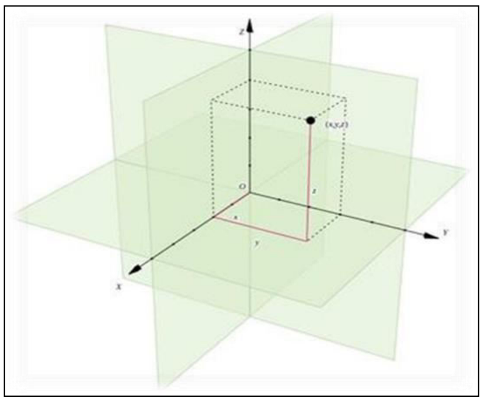


Figure 1: Describing point (1,1,1) [18]

In the 3D engine, moving an object in space causes vectors to be added or subtracted.

IV. PROBLEM STATEMENT

In the product development process, client input is essential for success. Mass customization has become critical in today's global market, with many buyers using 3D GUIs to modify or create products online. However, most online stores only allow customers to select product specifications, without updating the product's appearance to reflect their

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105037 |

choices. This passive approach may not work well for products where appearance plays a significant role in purchasing decisions. To address this issue, this study developed an online 3D product configurator consisting of 3D web viewing tools, an application server, and a backend system. The system allows customers to easily connect with products and customize their appearance, improving the online shopping experience.

V. THREEJS

Three JS is a JavaScript 3D library that allows users to create and showcase 3D visuals and animations on a web browser without requiring any browser plugins. It uses WebGL to accomplish this, making it functional on any browser that supports WebGL[19]. Three JS is advantageous in creating eye-catching 3D visuals without the need to master WebGL, a difficult technology that requires a thorough understanding of programming and writing shader code. The library was unveiled by Ricardo Cabello in April 2010, with over 650 contributors to date. Some of the project's most significant contributors are Branislav Ulicny, Paul Brunt, and Joshua Koo [20]. The code was initially written in ActionScript but was later converted to JavaScript by Cabello in 2009, with the renderer written as a distinct module, allowing for easy switching.

WebGL is a powerful tool that operates at a low level, making it less user-friendly than Three JS. It is an OpenGL ES 2.0-based technology that combines JavaScript and shader code, with the latter collaborating with the graphics processing unit while the former is used to control it. The result is shown as Document Object Model interfaces using the HTML5 canvas element. WebGL's current version is 2.0 [21].

A. Why Three JS?

Three JS was chosen for a project because it is designed for presenting 3D material and lacks game engine components, making it more compact. While it can be used for game development, it has basic tools for displaying 3D information. It has excellent documentation and import/export functionality, including the crucial JSON import feature. Babylon JS and Unity, both game engines, were also considered, but Unity was too heavy and Babylon JS, originally a Silverlight game engine, is now a JavaScript framework that handles game creation and 3D display on web browsers. The developers chose Three JS for the project because they had previously used it and were pleased with it.

Features

Many things that would be exceedingly challenging to create with WebGL are very effortless with Three JS. According to its developers, some of these features include the following [20]:

- Three.js is a JavaScript 3D library that allows users to create and display 3D visuals and animations in a web browser, without requiring any browser plugins.
- Three.js uses WebGL, which is supported by almost all modern web browsers, and does not require a deep understanding of the technology or shader code to create eye-catching 3D visuals.
- Three.js was created by Ricardo Cabello in 2010, and has since had over 650 contributors.
- Three.js is not a game engine, but rather a library that provides basic tools for displaying 3D information, making it compact and suitable for a variety of projects.
- Three.js has one of the best documentation systems and is well-developed, making it one of the most widely-used libraries of its kind.
- Babylon JS and Unity were considered as alternatives, but were not chosen because they are both game engines and too heavy for the project's needs.
- Three.js offers a range of renderers, cameras, lights, materials, objects, geometry, shaders, loaders, utilities, and export/import tools, making it a comprehensive solution for creating 3D visuals.
- Three.js also provides a substantial post-processing library, access to full WebGL capabilities, and a collection of temporal and three-dimensional arithmetic operations.
- Three.js has more than 150 files of code examples, support from a public forum and wiki, and a comprehensive API documentation.

Three JS is a popular JavaScript library for creating 3D graphics in web browsers. The library offers a variety of tools for creating and rendering scenes, including a powerful WebGL renderer [22], canvas renderer, and SVG renderer [19]. The scene is the foundation for building a 3D graphic display, and it records all the elements in it, including cameras, lights, and forms. Cameras come in two types: perspective and orthographic, and each has its own use case.

Simple animations can be achieved using Three JS, with users being able to choose between morph and skeleton animations. Lighting is also essential for illuminating the scene, with Five distinct types of lighting available from Three JS [20]. The materials assigned to a mesh strongly influence how a form appears, and the library provides various materials, including basic, depth, normal, face, lambert, and phong. Shaders are also available, and these are pieces of code that execute directly on the GPU, without taxing the CPU.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105037 |

The geometry of a 3D model stores all the information needed to depict it, and Three JS has a wide variety of forms available, including spheres, boxes, and planes. Custom forms can also be created by distributing all the geometry's vertices and faces. Finally, loaders and import and export tools enable users to include resources from outside the Three JS scenario. Supported formats include JSON, Obj, and several types of images and sceneries [24]. Several well-known 3D modelling programmes such as Blender or 3DS Max may also be used to import models by exporting them as JSON files.

B. Framework comparison

Three JS, Unity, and Babylon JS [3] are the three technologies that may be used to create WebGL content for browsers in this comparison. One of the most well-known game engines on the market, Unity, can create games for WebGL applications. An open-source WebGL-based 3D engine called Babylon JS was first developed by Microsoft engineers. A comparison for the most used framework can be seen in the (**table 1)** which show what are the added benefits and for what purpose each can be used.

Table 1: Framework comparison

Language	JavaScript	C#, UnityScript	JavaScript
License	MIT	Proprietary	Apache License 2.0
Mobile support	Yes	No	Yes
Development environment	No restrictions	Unity	No restrictions
Integrated physics	No	PhysX	Cannon JS, Oimo JS, Energy JS
Import	JSON, OBJ, FBX	OBJ, FBX	OBJ, FBX, Babylon, STL, JSON
Export	JSON, OBJ	No	Formats supported by Blender and 3dsMax
VR support	Yes	Yes	Yes

VI. Methodology Implemented

Basic Steps*:*

- Setting up a project, adding dependencies, locating and adding a 3D model, and using basic styles and markup
- Editor's component
- Scene component
- Project start.
- Install VS-Code

Project Setup:

- In the project, it combine's Three.js with MDB as the primary UI library and use MERN to develop 3D Configurator in React.
- The actions listed here are:
- According to the official guidelines, installing the required library and framework to the Editor i.e., VS-Code.

A. Create your 3D model by:

- In the /static root directory, create the /model folder.
- Locate a model in.gltf (JSON) or.glb (Binary) format shown in (**Figure 2**) (various loaders exist for each sort of format in Three.js; GLTFLoader is officially advised).
- Model may be downloaded and unzipped; the archive most likely includes a file with the extension.bin and the /textures folder.

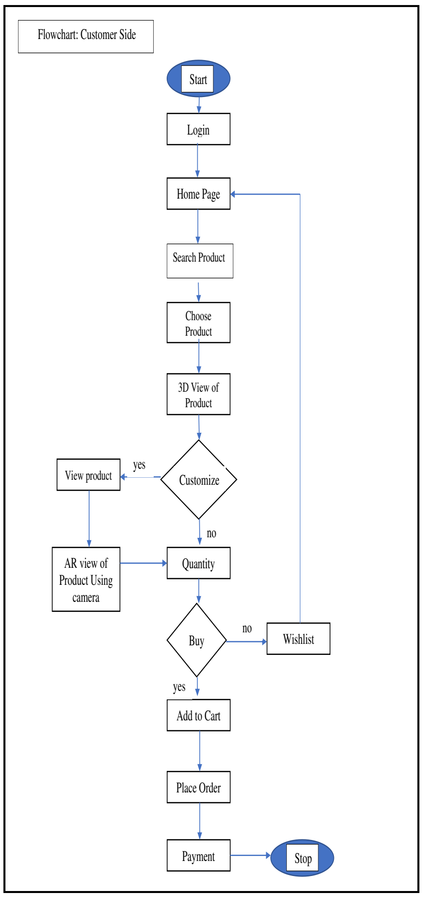
Element for canvas (canvas):

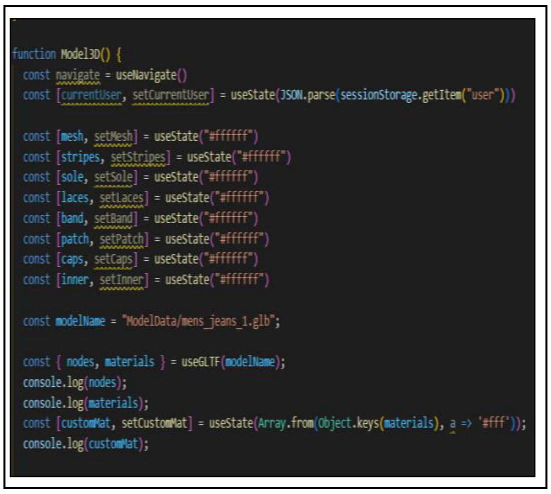
- This is the component that will replicate and apply a texture to in our 3D model.
- It has an ID, and the Scene component will require it.

B. Methods:

- Every time the editor canvas element changes, the function updateMesh() must be called. The scene component will process the MESH UPDATE event that transmit using this technique.
- applyColor() modifies the editor canvas's background colour (passed in the argument or random from the colorsMap array).

- The canvas element is given the scope of the parent container by the function onResize(), which is invoked when the window resizes. It is preferable to add storing their dimensions on window resize if you wish to add the ability to, say, insert text or shapes; this can only be done using this way.





```
function Model3D() {
  const navigate = useNavigate()
  const [currentUser, setCurrentUser] = useState(JSON.parse(sessionStorage.getItem("user")))

  const [mesh, setMesh] = useState("#ffffff")
  const [stripes, setStripes] = useState("#ffffff")
  const [sole, setSole] = useState("#ffffff")
  const [laces, setLaces] = useState("#ffffff")
  const [band, setBand] = useState("#ffffff")
  const [patch, setPatch] = useState("#ffffff")
  const [caps, setCaps] = useState("#ffffff")
  const [inner, setInner] = useState("#ffffff")

  const modelName = "ModelData/mens_jeans_1.glb";

  const { nodes, materials } = useGLTF(modelName);
  console.log(nodes);
  console.log(materials);
  const [customMat, setCustomMat] = useState(Array.from(Object.keys(materials), a => '#fff'));
  console.log(customMat);
```

Figure 2: Model .glb (Binary) format

Figure 2: Working model of the site

Flow Diagram:(Figure 3) is the easy depiction to understand how the working of the project development.

1) **Colorpicker:** Colorpicker is available right away and applying colour to the input event using the technique.
2) **Making a scene element:**
- Let's break down this level into 4 further actions:
- Adding Three.js to the project and utilising a class to initialise the Three.js scene
- Component Scene.Options.Vue
- Component of Scene.vue
3) **Setting up the Three.js Scene using a class:**
- The development of the primary components required for scene initialization will be separated out into a separate class for convenience's sake: a Scene.init.js file should be created in the /components/Scene/js folder.
- Scene, Camera, and Renderer are the three items we need to show anything using three.js in order to render the scene with the camera.
- Put everything into practise in the SceneInit class mentioned above:
- By executing the necessary methods in the init() function, initialise the scene, camera, and renderer in that order.
- Also include light, controls (OrbitControls, which are in charge of interfacing with the camera), and a a model loader (such as GPTLoader for.gltf models) .glb format).
- GLTFLoader and OrbitControls are imported from "three/examples/jsm/loaders/GLTFLoader" and "three/examples/jsm/controls/OrbitControls," respectively [3].

- After completing all of these processes, inject the rendered render element's HTML code into the component that called us.
- After setting up the scene's essential elements, it must now draw it on the screen. To do this, call update() method. Every time the screen is refreshed, the scene will be drawn again (for a standard screen, this is about 60 times per second).

4) Import:

- The SceneInit function is a tool for setting up the Three-scene.
- Find children's components in SceneOptions.
- ArraySibling is a tool for finding adjacent elements in an array.

5) Markup:

- It only have the SceneOptions component and vuetify-overlay in the markup. Scene is automatically added when the SceneInit () function is performed, which then calls the class with the same name.
- model— loaded model
- objects — An array of the loaded model's child elements. model objects (zone, object, mesh-all this implies the same thing).
- activeMesh — The current active object is activeMesh.
- isLoaded, showWireframes — Boolean flags for the model display mode and vuetify-overlay in wireframes (with or without wireframes) [25].
- editorCanvas — The editor component's canvas element, editorCanvas, will be used to generate a texture, new
- setActiveMesh() — Set the active object and give it a canvasTexture that is a duplicate of the canvas from the Editor component using the ActiveMesh() method. If it is a mousedown event, use a findArraySibling helper here (meaning it was called from the navigation button).
- createWireframe() — Create wireframes for a single model object toggle with createWireframe().
- toggleWireframes() — Use the.traverse() function to iterate through all model objects and set values for each wireframe's visibility in accordance with the current show Wireframes flag.

In order to implement scene and editor synchronisation:

- duplicating the Editor component canvas in a new texture.
- passing it as a mapoption for the active object's texture (mesh, zone).
- calling a texture update by signing up for an event called MESH UPDATE from the Editor component [23].

C.Finalization

- Developing and deploying our project is the final step will upload the static files.
- Configure the router further (router base with your repo-name)
- Run npm generate Install gh-pages from the terminal in your devDependecies
- Run any newly added package.json script.

VII. RESULT

The The creation of a web application for browsing products was the major goal of this. The viability of this sort of project for Three JSs is also assessed. To support this, evaluated Three JS's key features, presented some of its rivals, and contrasted Three JS with them.

A. ThreeJS evaluation

The capacity of the three JSs to provide results fast was tested because the majority of the project was completed in only a short frame. This type of prototyping showed how effectively the framework functions. When compared to Unity [26], for instance, there is far less discussion and documentation online, but the ThreeJS website offers a useful collection of examples of most of the functionality.

B. Product Display

At this point the user can sight the product from all angles and can make color changes to it. They need to use the arrow keys to highlight different portions and the color picker to adjust the color. Different Models are shown in (Figure 4&5).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105037 |

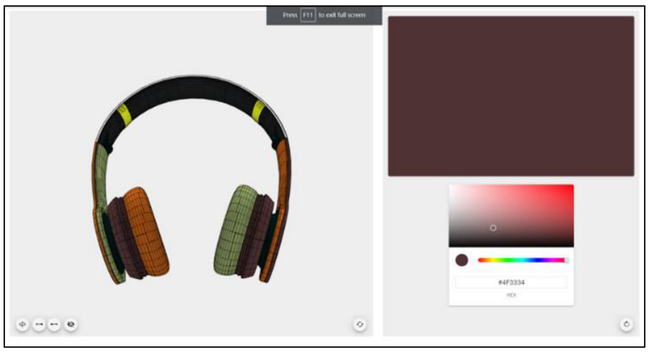


Figure 4: One face of the 3D model

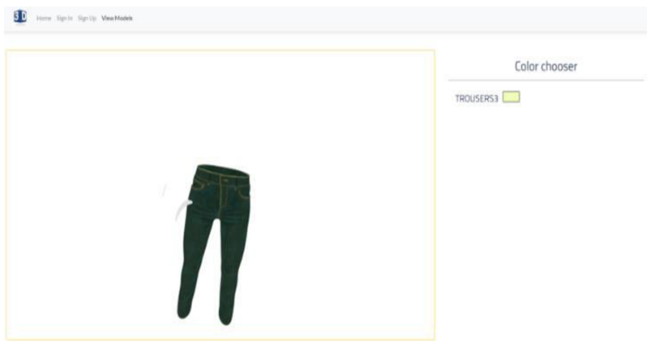


Figure 5: Model

VIII. CONCLUSION AND FUTURE SCOPE

CONCLUSION

In 2023, MERN is an excellent full-stack alternative. The front-end display stage (React.js), application stage (Express.js and Node.js), and database (MongoDB) stage all complete up its three-stage architecture pattern. It is perfect for creating a variety of apps. It provides quick, simple, cost-effective, and efficient app creation and deployment.

With addition it was able to implement Three.js and project 3D models into the web pages. These days it is difficult to find a page through which the use can customize the product and realize how it can look with different filters and color. Using bulky software to do this is not that feasible for a non-technical person. So, this website would simplify the work and hardware needs of the users.

The user will be able to use these replicas and generate similar real-life object with different printing services which are available on the internet. They can bring their product to life through the printing and designing services.

This website will also help reduce the paper requirements because the customizations will remain fixed and they could be edited any time.

FUTURE SCOPE

Adding more and more merchandise to the site so that there is diversity in the number of models of the merchandise and also be adding more customization options which will help the users do a lot more work efficiently.

In future uploading options would also be enabled for the users so that they can render their own models and use the free customizations available. Models saving feature will also be added according to the needs.

Addition of New features:

1) User Interface Design: This will involve creating a layout for the website, determining how the 3D merchandise views should be presented, and creating the UI in a user-friendly and user-friendly.
2) Integrating with AR with website for the view of merchandise.
3) Admin Portal for Management
4) Integration with Inventory and Ordering System: The configurator should be integrated with the company's inventory and ordering system to provide accurate pricing and availability information, and to facilitate the purchase of the customized merchandise.
5) Payment System Integration

REFERENCES

1. Unity. 2016 Getting started with WebGL development. https://docs.unity3d.com/Manual/webgl-gettingstarted.html
2. Three JS, 2017. Spotlight. https://threejs.org/docs/#api/lights/SpotLight
3. Hewitson, J. 2013. Three.js and Babylon.js: a Comparison of WebGL Frameworks. https://www.sitepoint.com/three-js-babylon-js-comparison-webgl-frameworks/
4. Altarteer, S., & Charissis, V. (2019). Technology acceptance model for 3D virtual reality system in luxury brands online stores. IEEE Access: Practical Innovations, Open Solutions, 7, 64053–64062. https://doi.org/10.1109/access.2019.2916353

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105037 |

5. Dai, K., Li, Y., Han, J., Lu, X., & Zhang, S. (2006). An interactive web system for integrated three-dimensional customization. Computers in Industry, 57(8–9), 827–837. https://doi.org/10.1016/j.compind.2006.04.017
6. Massaro, A., Vitti, V., Mustich, A., & Galiano, A. (2019). Intelligent real-time 3D configuration platform for customizing E-commerce products. International Journal of Computer Graphics & Animation, 9(4), 13–28. https://doi.org/10.5121/ijcga.2019.9402
7. Nyamsuren, P., Lee, S.-H., Hwang, H.-T., & Kim, T.-J. (2015). A web-based collaborative framework for facilitating decision making on a 3D design developing process. Journal of Computational Design and Engineering, 2(3), 148–156. https://doi.org/10.1016/j.jcde.2015.02.001
8. Potenziani, M., Callieri, M., Dellepiane, M., Corsini, M., Ponchio, F., & Scopigno, R. (2015). 3DHOP: 3D heritage online presenter. Computers & Graphics, 52, 129–141. https://doi.org/10.1016/j.cag.2015.07.001
9. Scopigno, R., Callieri, M., Dellepiane, M., Ponchio, F., & Potenziani, M. (2017). Delivering and using 3D models on the web: are we ready? Virtual Archaeology Review, 8(17), 1. https://doi.org/10.4995/var.2017.6405
10. Shojaei, D., Olfat, H., Rajabifard, A., & Briffa, M. (2018). Design and development of a 3D digital cadastre visualization prototype. ISPRS International Journal of Geo-Information, 7(10), 384. https://doi.org/10.3390/ijgi7100384
11. Zhao, H., McLoughlin, L., Adzhiev, V., & Pasko, A. (2018). 3D mass customization toolkits design, part I: Survey and an evaluation model. Computer-Aided Design and Applications, 16(2), 204–222. https://doi.org/10.14733/cadaps.2019.204-222
12. (N.d.). Researchgate.net. Retrieved February 2, 2023, from https://www.researchgate.net/profile/Chih-Hsing-Chu/publication/26502633_Online_Product_Configuration_in_e-Commerce_with_3D_Web_Viewing_Technology/links/5498b6c70cf2c5a7e342c545/Online-Product-Configuration-in-e-Commerce-with-3D-Web-Viewing-Technology.pdf
13. Koster, L., & Peters, R. (2020). CAD to VR. A guide to the conversion process.https://leonkoster.dev/Documents/CADXRPaper.pdf
14. Tu:, M. P. S. U. (n.d.). Lean Startup orientation: Empirical evidence on venture success. Utwente.Nl. Retrieved February 2, 2023, from https://essay.utwente.nl/74732/1/Schwery_BA_BMS.pdf
15. Dai, K.-Y., Li, Y.-S., Zhan, S.-S., & Xu, X.-Q. (2004). Three-dimensional online customization ordering system. 8th International Conference on Computer Supported Cooperative Work in Design, 2, 588-593 Vol.2 https://ieeexplore.ieee.org/abstract/document/1349259
16. Manferdini, A. M., & Remondino, F. (2010). Reality-based 3D modeling, segmentation and web-based visualization. In Digital Heritage (pp. 110–124). Springer Berlin Heidelberg.
17. Slick, J. (2011, June 23). What is 3D rendering in the CG pipeline? Lifewire. https://www.lifewire.com/what-is-rendering-1954
18. Sloka-Frey, K. 2013. Let's Build a 3D Graphics Engine: Points, Vectors, and Basic Concepts. Game Development Envato Tuts+; Envato Tuts.https://gamedevelopment.tutsplus.com/tutorials/lets-build-a-3d-graphics-engine-points-vectors- and-basic-concepts--gamedev-8143
19. Pettit, N. 2013. Beginner's Guide to three.js. http://blog.teamtreehouse.com/the-beginners-guide- to-three-js
20. Cabello, R. 2012. ThreeJS history. https://github.com/mrdoob/three.js/issues/1960
21. WebGL 2.0 quick reference. (n.d.). Lulu. Retrieved October 31, 2022, from https://www.lulu.com/shop/khronos-group/webgl-20-quick-reference/paperback/product-1vjrezz8.html?page=1&pageSize=4
22. W3 Schools. 2016. Differences between SVG and Canvas. http://www.w3schools.com/html/html5_svg.asp.
23. Cabello, R. 2017. Three.js Features. https://github.com/mrdoob/three.js/wiki/Features.
24. Petitcolas, J. 2015. Importing a Modeled Mesh From Blender to Three.js. https://www.jonathan-petitcolas.com/2015/07/27/importing-blender-modelized-mesh-in-threejs.html
25. Dirksen, J. 2013. Learning Three.js: The JavaScript 3D Library for WebGL. Birmingham: Packt Publishing.
26. Jarvis, M. 2015. Unity 5.3 makes WebGL support official. http://www.develop- online.net/news/unity-5-3-makes-webgl-support-official/021

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING