

Anjuman-I-Islam's
M.H.Saboo Siddik Polytechnic
8, M.H.Saboo Siddik Polytechnic Road, Mumbai 400008



FINAL YEAR DIPLOMA IN COMPUTER ENGINEERING
(2023-2024)

PROJECT REPORT ON
3D OUTFIT CUSTOMIZER: DESIGN YOUR OWN CUSTOM OUTFITS ONLINE

BY
220486 SHAIKH MOHAMMED HUSSAIN

UNDER THE GUIDANCE OF
PROF. ZAIBUNNISSA MALIK & PROF. MOHAMMED ALI



Maharashtra State Board of Technical Education (MS-BTE)
Mumbai (Autonomous) (ISO 9001:2008) (ISO/IEC 27001:2005)



CERTIFICATE

This is to certify that the project has been duly signed
and performed by

220486 – HUSSAIN SHAIKH

of Third year diploma in computer engineering during the academic year
2023-24. They have completed the project as per the guidance
prescribed by MSBTE

HEAD OF DEPARTMENT:

DATE: _____

INTERNAL EXAMINER:

DATE: _____

SIGN OF THE GUIDE:

DATE: _____

EXTERNAL EXAMINER:

DATE: _____

PRINCIPAL _____

DATE: _____

Acknowledgement

It is our esteemed pleasure to present the project report on “**SnX (3D Outfit Customizer)**”

We would firstly like to thank our Principal (I/c), Head of the Department & Guide Prof. Zaibunnisa Malik for encouraging and motivating us with her guidance and total support for our work. We shall also like to thank Prof. Mohammed Ali for working as our sub guide and making our path to integrity much simpler.

We also thank all the teachers who constantly motivated us and provided us their precious knowledge about the procedures carried out for making a project along with technical knowledge they have availed.

We would also like to thank our principal Mr. A.K Qureshi for providing us this Opportunity of integrating our own project and constantly supporting us throughout the process.

It would also be pleasure thanking all the staff, be it teaching or non-teaching who always understood by us and never made any problem tread our way.



Government Polytechnic, Thane

INFORMATION TECHNOLOGY DEPARTMENT

INFINITY-Forum

Presents

"TECHFUSION" - STATE LEVEL COMPETITION

This is to certify that Shaikh Mohammad Hussain
student of M. H. Saboo Siddik Polytechnic has awarded
as Winner in Event: Project Competition of State
Level Competition **"TECHFUSION"** organized on dated
18/03/2024 by Information Technology department.

Mr. J. R. Nikhade
Head of Info. Tech.

Dr. D. R. Mahajan
Principal

Table of Content

Sr. No	Chapter	Page No.
1.	Introduction and Background 1.1. Introduction 1.2. Background 1.3. Motivation 1.4. Problem Statement 1.4.1 Problem Definition 1.4.2 Proposed Work 1.4.3 Proposed Methodology 1.5. Objective and Scope 1.5.1 Scope 1.5.2 Objective 1.6. Advantages 1.7. Disadvantages	
2.	Literature Survey 2.1. Introduction 2.2. Research Papers 2.3. References 2.4. Conclusion	

Sr. No	Chapter	Page No.
3.	System Analysis 3.1. Introduction 3.2 System Design 3.2.1. Introduction to System Planning 3.2.2. Software Design Approach 3.3. Gantt Chart 3.4. Time Line Chart 3.5. Cost Estimation 3.5.1 Cost Beneficial Analysis 3.6. Feasibility 3.7. Conclusion	
4.	System Design 4.1. Introduction 4.2 Block Diagram 4.3. System Architecture 4.4. Data Flow Diagram 4.5. Table Structure 4.6. State Transition Diagram 4.7. E-R Diagram 4.8. Conclusion	
5.	Implementation 5.1. Introduction 5.2. Algorithm 5.3. Flowchart 5.4. Coding 5.5. Conclusion	

Sr. No	Chapter	Page No.
6.	Testing and Debugging 6.1. Introduction 6.2. Testing Approach 6.3. Test Plan 6.4. Debugging Approach 6.5. Test Cases 6.6. Conclusion	
7.	Snapshots 7.1. Snapshots	
8.	Future Development and Conclusion 8.1. Introduction 8.2. Limitations 8.3. Future Enhancement 8.4. Reference and Bibliography	

Chapter 1: Introduction and Background

Content:

- 1.1. Introduction
- 1.2. Background
- 1.3. Motivation
- 1.4. Problem Statement
 - 1.4.1. Problem Definition
 - 1.4.2. Proposed Work
 - 1.4.3. Proposed Methodology
- 1.5. Objective and Scope
 - 1.5.1. Scope
 - 1.5.2. Objectives
- 1.6. Advantages
- 1.7. Disadvantages

1.1. Introduction

This section delves into the background, motivation, problem statement, objectives, scope, advantages, disadvantages, and limitations of the 3D Outfit Customizer project. It highlights the need for a more immersive and personalized fashion shopping experience that empowers users to express their individuality through 3D customization. The section outlines the project's objectives, including curating a vast collection of clothing items, enabling 3D design and personalization, providing an immersive platform, facilitating sharing and collaboration, offering a user-friendly interface, and making online fashion shopping enjoyable and personalized. It also identifies the project's advantages, such as enabling users to create unique outfits, providing realistic 3D previews, streamlining the shopping process, and enhancing the overall online shopping experience. The section acknowledges the technical challenges, limitations in fabric texture representation, and website performance complexity that may arise during development. It also mentions the limited availability of fabric textures and patterns, potential compatibility issues, and privacy concerns that may need to be addressed.

1.2. Background

Traditional offline outfit shopping methods do not align with modern preferences, and even conventional online shopping falls short in providing a comprehensive solution for users to fully express their style. The fashion industry lacks a platform that allows diverse preferences to flourish through 3D customization.

1.3. Motivation

The motivation behind the "3D Outfit Customizer" project is to bridge the gap between traditional online fashion shopping and the desires of modern consumers. The project seeks to leverage 3D technology and A to provide users with an immersive, interactive, and personalized shopping experience that empowers them to design their own outfits, thereby fostering a sense of creativity and individuality in the fashion industry.

1.4 Problem Statement

1.4.1 Problem Definition

Many online shopping platforms lack the depth and versatility required to cater to the diverse and ever-evolving fashion preferences of today's consumers. While they may offer a wide range of products, the ability to tailor and customize these items according to one's unique taste is often limited. Users are typically constrained to selecting from pre-designed options, which restricts their creative expression and can result in a generic shopping experience. This limitation hinders the full realization of personal style and makes it

challenging for individuals to find and create outfits that truly resonate with their fashion sensibilities.

Furthermore, the absence of a comprehensive 3D customization solution within the fashion industry exacerbates this problem. The conventional 2D representation of clothing on online platforms fails to capture the subtleties of fit, style, and individual preferences, leaving a gap in the market for a more immersive and interactive solution. By introducing a 3D outfit customizer, this project aims to bridge this gap and empower users to explore and materialize their unique fashion visions, ultimately revolutionizing the way fashion is both experienced and consumed in the digital age.

1.4.2 Proposed Work

To conduct stream analysis for the "SNX 3D Outfit Customizer" project, several steps are followed:

- **Define Objectives:** Establish specific goals for the analysis, focusing on enhancing user engagement and personalization within the 3D outfit customization experience.
- **Data Gathering:** Collect relevant data from users, including preferences, body measurements, style choices, past selections, and any other pertinent information.
- **Data Preprocessing:** Ensure the collected data is clean and consistent, handling any missing values or inconsistencies. This might involve organizing and structuring data for efficient use in 3D design.
- **Feature Selection/Extraction:** Identify and extract key features from the data essential for customizing the 3D outfit experience. This could include style preferences, body measurements, preferred colors, and design choices.
- **3D Model Design:** Utilize appropriate 3D modeling techniques and tools, such as React Three Fiber and Three.js, to design the customizable outfit environment within the SnX platform.
- **Model Training:** Develop and refine the 3D outfit customization models using the selected technologies, ensuring they accurately represent user preferences and choices.
- **Model Evaluation:** Assess the performance and usability of the 3D outfit customization models within the SnX platform, ensuring they meet user expectations and provide an engaging experience.
- **Model Refinement:** Fine-tune the 3D outfit customization models based on user feedback and usage data, optimizing them for improved personalization and satisfaction.

- **Integration and Deployment:** Integrate the customized 3D outfit models seamlessly into the SnX platform, utilizing ReactJS and MongoDB for efficient storage and retrieval of user data.
- **Monitoring and Maintenance:** Continuously monitor the performance of the 3D outfit customization features within the SnX platform, making updates and improvements as needed to ensure a seamless and engaging user experience.

1.4.3 Proposed Methodology

In the proposed methodology for the SnX 3D Outfit Customizer, we focus on creating an immersive and efficient user experience through the integration of various technologies such as ReactJS, Next.js, MongoDB, React Three Fiber, and Three.js. Here's how the methodology translates:

- **Enhanced User Experience:** The incorporation of ReactJS and Next.js ensures a seamless and interactive user interface, facilitating easy navigation and engagement within the 3D outfit customization platform.
- **Efficient Data Management with MongoDB:** MongoDB serves as the backbone for storing and managing user data, including preferences, style choices, and past selections, ensuring fast access and retrieval of information.
- **3D Modeling and Visualization with React Three Fiber and Three.js:** Leveraging React Three Fiber and Three.js, we create a visually stunning and immersive 3D environment for outfit customization. Users can interact with lifelike representations of clothing items and accessories, enhancing their overall experience.
- **Answer Validation and Personality Assessment:** The system validates user responses by matching them with predefined answers stored in the database. Additionally, marking criteria based on Holland's code career model are utilized for evaluating user responses and suggesting personalized outfit recommendations.
- **Personality Profiling and Result Storage:** After users answer all questions, the system utilizes Holland's code career model to suggest the top 3 personality profiles. User results are stored in MongoDB for future reference, allowing users to view and track their personalized outfit recommendations accordingly.

In summary, the proposed methodology for the SnX 3D Outfit Customizer leverages advanced technologies such as ReactJS, Next.js, MongoDB, React Three Fiber, and Three.js to deliver an immersive, efficient, and personalized outfit customization experience for users.

1.5. Objective & Scope

❖ Objective

- To study the 3D Outfit Customizer Systems
- To curate a vast collection of clothing items, fabrics, styles, patterns, and accessories for users to choose from and customize.
- To enable users to design and personalize outfits in a 3D virtual environment, fostering a sense of creativity and individuality in their fashion choices.
- To provide users with an immersive and interactive platform that enhances their shopping experience by allowing them to express their unique style.
- To offer a user-friendly interface accessible on various devices, making the customization process intuitive and accessible.

❖ Scope

- A vast collection of clothing items, fabrics, styles, patterns, and accessories for users to choose from and customize.
- A 3D virtual design studio for outfit creation and customization.
- A vast collection of clothing items, fabrics, styles, patterns, and accessories to choose from.
- Interactive elements to adjust outfit details such as sleeves, collars, and hemlines.
- Realistic 3D previews to visualize designs.
- A user-friendly interface accessible on various devices.

1.6. Advantages

- Empowering users to create outfits that reflect their unique style.
- Providing realistic 3D models for accurate design visualization.
- Streamlining the shopping process by enabling the creation and purchase of custom outfits in one place.
- Offering an enjoyable and interactive online shopping experience.

1.7. Disadvantages

- Technical challenges related to achieving realistic 3D rendering.
- Limitations in accurately representing fabric textures in the 3D model.
- Complexity in maintaining website performance.

Chapter 2: Literature Survey

Content:

- 2.1. Introduction
- 2.2. Research Papers
- 2.3. References
- 2.4. Conclusion

2.1. Introduction

The literature survey will delve into the history and evolution of online fashion shopping, emphasizing the limitations of traditional approaches in enabling users to fully express their unique style. We will examine existing 3D virtual design studios and fashion platforms to gain insights into the techniques and technologies that have been employed in similar contexts. In this survey, we will read a number of IEEE, IJERT, Springer Research Papers. This survey will provide a foundation for identifying best practices, potential challenges, and opportunities for innovation in the "3D Outfit Customizer" project, ensuring that it builds upon and surpasses the current state of the art in the field. This survey will help define the problem statement.

2.2. Research Papers

Paper Title 1: Dual-Mode User Interfaces for Web-Based Interactive 3D Virtual Environments Using Three.js

Author: Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo

Published in: 2017

Abstract: Dual-mode user interface (2D+3D) enables faster information retrieval and higher user satisfaction than 3D websites alone. 2D interface has the best UI quality, 3D interface is the most aesthetically pleasing. 3D interface has worse text presentation than 2D and dual-mode interfaces. Dual-mode interface lacks a content search function, which could slow down data retrieval for users used to Ctrl+F. Dual-mode user interfaces have potential to enhance user interaction in 3D virtual environments. Future work should integrate 3D and 2D views more closely and expand the scope of the study.

DOI: https://doi.org/10.1007/978-3-319-68059-0_47

Paper Title 2: React JS – An Emerging Frontend JavaScript Library

Author: Pratik Sharad Maratkar and Pratibha Adkar

Published in: 2021

Abstract: React JS is an emerging and popular frontend JavaScript library known for its fully component-based architecture. It simplifies the development of rich UIs by using reusable components. Facebook, Instagram, and other organizations back React JS and use it in their web applications. React JS offers tools for easy debugging, including Chrome extensions. React primarily deals with the View part in MVC, so other tools are needed for backend development. Some developers may find JSX programming challenging during the learning phase. React's environment evolves rapidly, requiring developers to stay updated with its changes. The paper provides a comparison between React JS and Angular JS, highlighting their differences in terms of development, performance, and usage. It discusses React JS's architecture, including React Virtual DOM, one-way data flow, React components, and JSX syntax

DOI: <https://www.irejournals.com/paper-details/1702778>

Paper Title 3: Frontend Development with React.js

Author: Anjali Rananavare

Published in: 2022

Abstract: React.js is a popular choice for building UIs with a virtual DOM, one-way data flow, and component-based architecture. It is lighter weight, faster, and easier to use than other frameworks. It requires importing libraries for state and model management and moves away from class-based modules, but it simplifies frontend development and is popular in the industry.

DOI: <https://www.irjet.net/archives/V7/i6/IRJET-V7I61149.pdf>

Paper Title 4: React Apps with Server-Side Rendering: Next.js

Authors: Harish A Jartarghar, Girish Rao Salanke, Ashok Kumar A.R, Sharvani G.S, Shivakumar Dalali

Published in: 2022

Abstract: The paper discusses the use of React.js and Next.js to develop web applications with a focus on server-side rendering. It highlights the problem of client-side rendering causing slow page loading and proposes the use of Next.js, a lightweight React framework, to address this issue. Next.js leverages server-side rendering, which allows the server to process web pages, fetch user-specific data, and send it to the browser over the internet. The paper also emphasizes Next.js' benefits for Search Engine Optimization (SEO) and its automatic routing system using the "pages" directory. Next.js is a lightweight React framework that uses server-side rendering to improve page loading speed and performance. It also has benefits for Search Engine Optimization (SEO) and an automatic routing system. Next.js is a good solution for developing web applications where fast page loading and SEO are important considerations.

DOI: https://www.researchgate.net/publication/347242422_Diving_into_Threejs

Paper Title 5: React JS (Open Source JavaScript Library)

Authors: Alok Kumar Srivastava, Vaishnavi Laxmi, Payal Singh, Km Pratima, Vibha Kirti

Published in: 2022

Abstract: The paper discusses React JS, an open-source JavaScript library for building user interfaces. React JS is used for handling the view layer in single-page applications and mobile application development. It highlights some of the key features of React JS, such as JSX, stateful components, and the Virtual Document Object Model (Virtual DOM). The paper also explains how React JS can be integrated into various types of applications, both simple and complex. The paper discusses the benefits of using React JS for front-end web application development. It emphasizes React's ability to create large and complex web applications that can update data without requiring page refreshes. React JS is a popular and powerful JavaScript library for front-end web development. It has a wide range of features and benefits, making it a good choice for developing both simple and complex web applications.

DOI: https://ijirt.org/master/publishedpaper/IJIRT153854_PAPER.pdf

Paper Title 6: Modern Web-Development using React.js

Authors: Bhupati Venkat Sai Indla and Yogeshch

Published in: 2018

Abstract: ReactJS: JavaScript library for building modular UIs with virtual DOM, unidirectional data flow, and easy learning curve. Limitations: Primarily handles View entity in MVC, inline templates and JSX may be complex, failures at compile time. Overall: Modern and efficient approach to web development, especially for dynamic and interactive user interfaces. The development process, offers high performance through virtual DOM, and encourages unidirectional data flow. The paper emphasizes

ReactJS's potential to impact the way web applications are developed and its ability to meet the demands of modern web development trends."

DOI: <https://iarjset.com/wp-content/uploads/2021/03/IARJSET.2021.8229.pdf>

Paper Title 7: Review on React JS

Author: Dimpy Bansal

Published in: 2020

Abstract: React JS is an open-source JavaScript library for building user interfaces. It is one of the most popular front-end development frameworks and is used by companies like Facebook, Instagram, and Airbnb. React JS has a number of key features that make it a popular choice for developers. It is declarative, meaning that developers can describe the desired state of the UI and React will efficiently update the DOM to match that state. React JS also uses a component-based architecture, which makes it easy to create reusable and maintainable code. Additionally, React JS has a virtual DOM, which improves performance by only updating the parts of the DOM that have changed. React JS can be used to build a variety of web applications, including single-page applications (SPAs), mobile apps using React Native, and e-commerce websites.

DOI: <https://www.ijcrt.org/papers/IJCRT2004607.pdf>

Paper Title 8: Role of Node.js in Modern Web Application Development

Authors: Ghansham Jadhav and Flavia Gonsalves

Published in: 2020

Abstract: The paper discusses the role of Node.js in modern web application development. It explains Node.js's modularity, its built-in package manager (Node Package Manager or NPM), and its working architecture. The key features of Node.js, including event-driven I/O, single-threaded, and asynchronous programming, are discussed with examples to provide insights into Node.js's working architecture. "Node.js is a JavaScript runtime environment built on Chrome's V8 JavaScript engine, designed for server-side applications. It focuses on low memory consumption and performance, making it suitable for building scalable and lightweight applications. Node.js allows developers to use JavaScript for both client and server-side scripting. It uses event-driven, non-blocking, and asynchronous approaches, enabling it to handle concurrent requests efficiently.

DOI: <https://www.scribd.com/document/597752129/RESEARCH-PAPER-Node-js>

Paper Title 9: Comprehensive Analysis of React-Redux Hybrid App Development Framework

Authors: Shravan G V, Prof. Anitha Sandeep

Published in: 2020

Abstract: The research paper explores the use of the React-Redux framework for developing hybrid web applications that can run on both Windows and iOS platforms. React-Redux is a framework that allows developers to create applications for multiple native platforms using a single codebase written in JavaScript ES6. The framework utilizes Redux for state management in React Native applications and incorporates various dependencies developed by other React Native developers, such as database and user interface components. The research aims to judge the user experience of web applications developed using the React-Redux framework as satisfactory. The paper suggests that creating user interfaces with React-Redux is

easier compared to other platforms. React-Redux is capable of generating both simple and complex applications for cross-platform use, with a focus on high data fetching without caching.

DOI: https://hpi.de/fileadmin/user_upload/fachgebiete/doellner/publications/2012/DHK2012/paper.pdf

Paper Title 10: Survey And Analysis Of Rendering Realtime 3D Object On Cross-Browser & Cross-Platform Using WebGL

Authors: Yogiraj Patil, Kirti Wanjale

Published in: 2020

Abstract: The paper focuses on rendering real-time 3D objects using WebGL, a Web Graphics Library, and JavaScript. It discusses the challenges of rendering 3D objects in real-time, considering factors like varying screen sizes and processing power of different devices. It discusses the use of JavaScript frameworks for developing 3D web applications. The paper explores various rendering approaches and techniques for complex 3D objects in real-time using web browsers and WebGL. It emphasizes the importance of optimized rendering for web-based 3D applications. The paper mentions the use of glTF (GL Transmission Format) for efficient 3D model representation in WebGL. The paper acknowledges the challenges of rendering 3D objects in real-time on various devices and screen sizes. It highlights the role of JavaScript frameworks and WebGL in enabling 3D web applications. The paper mentions the importance of efficient data representation using technologies like glTF.

DOI: <https://www.jetir.org/papers/JETIR2105815.pdf>

Paper Title 11: Robust Real-Time Shadows for Dynamic 3D Scenes on the Web

Authors: Tim Nicolas Eicke, Yvonne Jung, and Arjan Kuijper

Published in: 2014

Abstract: The paper addresses the challenge of rendering high-quality shadows in 3D scenes on the web using the open-source JavaScript framework X3DOM, which integrates declarative 3D into HTML5. The authors examine existing shadow mapping techniques and develop a concept for enhancing shadow display in X3DOM. They implement Variance Shadow Maps and Parallel Split Shadow Maps (PSSM) to improve shadow quality and discuss the limitations of WebGL, the graphics library on which X3DOM is based, in comparison to other 3D frameworks like Three.js. The research focuses on improving the quality of real-time shadows in web-based 3D scenes.

DOI: https://doi.org/10.1007/978-3-319-07857-1_101.

Paper Title 12: Movie Data Visualization Based on WebGL

Authors: Min Li, Chunfang Li

Published in: 20201

Abstract: The paper uses WebGL, a technology that allows for the rendering of interactive 3D graphics on web pages without using plugins. It utilizes the open-source framework Three.js to create complex 3D scenes. The research implements five types of 3D data visualization charts: histograms, pie charts, maps, earth representations, and force-directed graphs. These visualizations are used to display film-related data. The paper mentions that 3D data visualization charts are rarely involved in current data visualization practices, but it does not elaborate on the specific limitations or disadvantages of 3D data visualization compared to 2D visualization.

DOI: 10.1109/SNPDWinter52325.2021.00023.

Paper Title 13: Comprehensive Analysis of React-Redux Development Framework

Authors: Shravan G V and Prof. Anitha Sandeep

Published in: 2020

Abstract: The research paper discusses the utilization of the React-Redux framework for developing hybrid applications compatible with both Windows and iOS platforms. The framework leverages JavaScript ES6 for implementation. React-Redux is a combination of React, developed by Facebook, and Redux for state management in React Native. The paper also mentions the use of other dependencies developed by React Native developers, such as databases and user interfaces. "The research work aims to judge the user experience of web applications created using the React-Redux framework as satisfactory. It examines whether user interface creation is easier compared to other platforms. It differentiates between the generation of simple and complex applications using React-Redux.

DOI: <https://core.ac.uk/download/pdf/301384737.pdf>

Paper Title 14: Server-Based Rendering of Large 3D Scenes for Mobile Devices Using G-Buffer Cube Maps

Authors: Juergen Doellne, Benjamin Hagedorn

Published in: 2020

Abstract: The paper introduces a server-client approach for rendering large 3D scenes on mobile devices. Instead of streaming 3D scene data to clients, the approach splits 3D rendering into two processes: a server process for real-time rendering of virtual panoramas represented by G-buffer cube maps and a client process for reconstructing the 3D scene and enabling user interaction. The paper demonstrates that this server-based rendering approach is effective in rendering large 3D scenes on mobile devices. It decouples the complexity of the 3D scene from data transmission complexity, allows for advanced 3D rendering on the server, and provides a high degree of protection for 3D content while supporting interactive user experiences on clients. The paper does not explicitly mention any disadvantages or gaps in the research. Different types of 3D visualizations are implemented for film-related data, such as histograms, pie charts, maps, and force-directed graphs.

DOI: <https://ijcrt.org/papers/IJCRT2205332.pdf>

Paper Title 15: The Research and Design Of 3D Web Guide System Based On WebGL

Authors: Cui Peng

Published in: 2021

Abstract: The paper presents a 3D Web guide system based on WebGL and three.js. It involves the use of HTML5 and Canvas for rendering, three.js for scene creation and rendering, NodeJS for server-side implementation, Socket.IO for WebSocket communication, and A* algorithm for shortest path search. The research paper describes the development of a 3D Web guide system that allows users to navigate unfamiliar environments using their mobile phones. It employs WebGL, three.js, and various web technologies to create interactive 3D scenes and calculate optimal paths between scenes. The system is designed for use in large amusement parks and shopping malls, serving as a form of advertising and a new type of 3D web application.

DOI: <https://www.ijtsrd.com/papers/ijtsrd42490.pdf>

Paper Title 16: Performance Optimization using MERN stack on Web Application**Authors:** Sourabh Mahadev Malewade, Archana Ekbot**Published in:** 2021

Abstract: The research project focuses on the development of an e-commerce web application using the MERN stack, which includes React.js, MongoDB, Node.js, and Express.js. The authors discuss the methodologies and technologies used to create and implement the web application. The use of React.js, MongoDB, Node.js, and Express.js in building the web application. The advantages of using Node.js for asynchronous, event-driven programming. The role of Express.js in simplifying back-end code and providing middleware support. The benefits of React.js in building user interfaces with components.

DOI: <http://ijrra.net/Vol5issue1/IJRR-05-01-27.pdf>**Paper Title 17: Efficient visualization of 3D models by web browser****Authors:** Bartosz Sawicki and Bartosz Chaber**Published in:** 2013

Abstract: The paper presents a software module designed for efficient and convenient visualization of 3D models inside a web browser environment. It is written purely in JavaScript and takes advantage of the new HTML5 standard. The authors focus on mobile devices, emphasizing efficiency and low network usage. They propose a solution based on progressive mesh streaming and compare it with server-side rendering approaches. The findings of the paper include the successful development of a JavaScript-based web component for 3D model visualization in web browsers.

DOI: <https://www.jetir.org/papers/JETIR2105815.pdf>**Paper Title 18: 3D Rubik's Cube - Online 3D Modeling System Based on WebGL****Authors:** Buyun Sheng, Feiyu Zhao, Chenglei Zhang, Xiyan Yin, Yao Shu**Published in:** 2017

Abstract: The paper introduces an online 3D modeling system called "3D Rubik's Cube" based on WebGL and Three.js. It achieves online 3D modeling through web front-end technologies and the establishment of a web server using Node.js. The system enriches 3D modeling functionality and enhances rendering effects through an improved Phong reflection model, Constructive Solid Geometry (CSG) tree, and triangular patches intersection testing and division algorithms. The paper demonstrates the development of an online 3D modeling system that allows cloud-based 3D model design. It leverages WebGL for 3D rendering and provides a range of 3D modeling functions.

DOI: 10.1109/ITNEC.2017.8284798.**Paper Title 19: WEBAPP SERVICE FOR BOOKING****Authors:** Saundariya K, Prabakaran D**Published in:** 2021

Abstract: The paper describes the development of a web application for booking handyman services. The system is built using the MERN stack, which includes MongoDB for the database, Express JS for the server, React JS for the front-end, and Node JS for the back-end. The system uses technologies like Axios for HTTP requests, JWT for data security and authentication, React-Redux for global state management, and Node-Mailer for sending notifications through email. The research presents a user-friendly website that

allows users to easily book handyman services online. It offers various services like cleaning, COVID-sanitization, furniture maintenance, electrical works, appliance repair, house painting, and plumbing. Users can select services, view available professionals based on location and cost, and book them. Handyman professionals can showcase their skills and accept or decline tasks.

DOI:

<https://books.google.com/books?hl=en&lr=&id=Xja9BwAAQBAJ&oi=fnd&pg=PP1&dq=three.js&ots=Rs3arD1d3b&sig=docKcu1uqZPFerJwohOOofpd-BZE>

Paper Title 20: HANDYMAN USING MONGO DB, EXPRESS JS, REACT JS, NODE JS

Authors: Abirami M, Srimathi B, Senthil Kumaran R, Nagarajan G (IEEE Member)

Published in: 2021

Abstract: The paper describes the development of a web application for booking handyman services. The system is built using the MERN stack, which includes MongoDB for the database, Express JS for the server, React JS for the front-end, and Node JS for the back-end. The system uses technologies like Axios for HTTP requests, JWT for data security and authentication, React-Redux for global state management, and Node-Mailer for sending notifications through email. The research presents a user-friendly website that allows users to easily book handyman services online. It offers various services like cleaning, COVID-sanitization, furniture maintenance, electrical works, appliance repair, house painting, and plumbing. Users can select services, view available professionals based on location and cost, and book them. Handyman professionals can showcase their skills and accept or decline tasks. The system aims to provide a convenient and cost-effective solution for connecting users with professional workers.

DOI: https://link.springer.com/chapter/10.1007/978-1-4302-3997-0_7

Paper Title 21: Immersive 3D Modeling with Blender and Off-the-Shelf Hardware

Authors: Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo

Published in: 2021

Abstract: The 3D modeling application was implemented in a short time frame using readily available hardware and software. The hardware used included a standard desktop computer, a 3D display (Panasonic TV), PlayStation Move controllers for user input, and a head-mounted PS Move controller for head tracking. The application combined 3D modeling tools with logical mappings to controller buttons, making it more intuitive for users.

DOI: 10.1109/3dui.2013.6550243

Paper Title 22: A Framework for Browser-based Multiplayer Online Games using WebGL and WebSocket

Authors: Bijin Chen, Zhiqi Xu

Published in: 2011

Abstract: The paper introduces a framework for creating browser-based multiplayer online games using HTML5, WebGL, and WebSocket technologies. It discusses the background of these technologies, such as HTML5's support for multimedia and graphics, WebGL for 3D graphics, and WebSocket for real-time communication. The authors implement a framework for a multiplayer online game (MOG) using Three.js for 3D rendering and jWebSocket for communication. The architecture of the framework involves clients, web servers, and game servers, with web workers optimizing communication between servers and clients.

DOI: [10.1109/ICMT.2011.6001673](https://doi.org/10.1109/ICMT.2011.6001673).

Paper Title 23: Investigating Web3D topics on StackOverflow: a preliminary study of WebGL and Three.js

Authors: Farag Almansoury, Sègla Kpodjedo, and Ghizlane El Boussaidi

Published in: 2020

Abstract: The paper investigates the attention and support received by WebGL and Three.js, two important Web3D technologies, on Stack Overflow. The authors focused on questions tagged with either WebGL or Three.js and extracted information from these questions, including the number of questions, cumulative views, and average views for each tag. They also analyzed the community support by looking at the failure rate (percentage of questions without accepted answers) and the median wait time for satisfactory answers.

DOI: [0.1145/3424616.3424726](https://doi.org/0.1145/3424616.3424726).

Paper Title 24: Web 2.0 and Virtual World Technologies: A Growing Impact on IS Education

Authors: Albert L. Harris and Alan Rea

Published in: 2009

Abstract: The paper discusses the use of Web 2.0 technologies and virtual world technologies in information systems (IS) education. It explores various Web 2.0 technologies, including wikis, blogs, podcasts, social networks, and virtual worlds. The authors examine how these technologies are being incorporated into IS education and discuss their potential advantages and disadvantages.

DOI: https://link.springer.com/chapter/10.1007/978-1-4302-3997-0_7

Paper Title 25: Research and Application of Web3D Exhibition Based on WebGL and Html5

Authors: M.J. Bian, J. Gao, H.H. Gao, J.P. Xu

Published in: 2015

Abstract: The paper presents a solution for creating Web3D exhibitions by combining WebGL and HTML5. It introduces the Web3D Exhibition Building System (Web3D-EBS), which is designed to build Web3D exhibitions in web applications. This system aims to improve compatibility and efficiency in Web3D exhibition applications, eliminating the need for plugins and relying on GPU rendering.

DOI: <https://link.springer.com/content/pdf/10.1007/s00607-012-0275>

Paper Title 26: REACT JS – AN FRONTEND JAVASCRIPT LIBRARY

Authors: Avinash Mishra Arshita Gupta

Published in: International Research Journal of Modernization in Engineering Technology and Science, Volume 04/Issue 11, November 2022

Abstract: The choice of the correct frontend framework or library is fundamental for software and application development. React.js plays a crucial role in frontend development, providing developers with opportunities to build new applications. This paper explores how React.js is instrumental in building applications, its merits, and key features. With over 80,000 sites and more than 1500 developers using React.js for web and mobile frontend development, it has become a key player in the industry. Companies like Amazon and PayPal utilize React.js for a significant portion of their application's frontend development. The paper analyzes the features of this library and its advantages over other frameworks.

React.js, as an open-source frontend JavaScript library, simplifies the development of user interface components

DOI:

https://www.irjmets.com/uploadedfiles/paper//issue_11_november_2022/31217/final/fin_irjmets1668401774.pdf

Paper Title 27: The usage of Vue JS framework for web application creation

Authors: Peter Pšenák, Matúš Tibensky

Published in: 2020

Abstract: The paper concentrates on the positive sides of front-end frameworks. It creates a small webpage that shows some of the basic capabilities of a JavaScript front-end framework called Vue JS and the easiness of its implementation. It explains the need for frameworks in development and how companies can benefit from them. Vue JS is considered user-friendly, making it a suitable tool for teaching web development, and it allows for incrementally adoptable web development. The paper provides a practical example of creating an interactive web app using Vue JS, demonstrating its simplicity and conciseness.

DOI: 10.35406/MI.2020.2.61

Paper Title 28: Review Paper on Web Frameworks, Databases, and Web Stacks

Authors: Jyoti Shetty, Deepika Dash, Akshaya Kumar Joish, Guruprasad C

Published in: 2020

Abstract: In today's world, online applications have become an essential component of various industries. The development of web applications involves identifying product requirements, design, coding, and testing using frameworks and technologies. Front-end and back-end frameworks, along with databases, play crucial roles in web development. This paper discusses the importance of choosing the right combination of these components, highlighting a few key frameworks, languages, databases, and web stacks.

DOI: <https://www.irjet.net/archives/V7/i4/IRJET-V7I41078.pdf>

Paper Title 29: React Apps with Server-Side Rendering: Next.js

Authors: Harish A Jartarghar, Girish Rao Salanke, Ashok Kumar A.R, Sharvani G.S, Shivakumar Dalali

Published in: 2022

Abstract: Web applications are developed using a variety of different web frameworks, and developers can pick from a wide range of web frameworks when developing a web application. React.js library provides flexibility for building reusable User Interface (UI) Components. It uses the approach of client-side rendering, which loads the HTML content using Javascript. The client-side rendering causes the page to load slowly and the client communicates with the server for run-time data only. Next.js Framework solves this problem by using server-side rendering. When the browser requests a web page, the server processes the web page by fetching the user's specific data and sending it back to the browser over the internet. Next.js helps the Search engines to crawl the site for better Search Engine Optimization (SEO).

DOI: <https://jtec.utem.edu.my/jtec/article/view/6192/4083>

Paper Title 30: Modern Front End Web Architectures with React.Js and Next.Js

Authors: Mochammad Fariz Syah Lazuardy and Dyah Anggraini

Published in: 2022

Abstract: This paper discusses the advantages and disadvantages of React.js and Next.js in the development of the State Civil Apparatus Information System (SIASN) web application. React.js is a JavaScript library for building user interfaces, while Next.js is a framework that enables server-side rendering with React concepts. The study highlights the steps involved in building the SIASN application's front-end and provides insights into the strengths and weaknesses of these technologies. React.js offers reusable components and supports both class and functional paradigms, making it versatile for front-end development. Additionally, it allows for client-side rendering and integrates with Redux for state management. On the other hand, Next.js facilitates server-side rendering, static site generation, and incremental static regeneration, enhancing SEO and application performance. By comparing these two technologies, the paper offers valuable insights for developers working on modern front-end web architectures.

DOI: <https://irjaes.com/wp-content/uploads/2022/02/IRJAES-V7N1P162Y22.pdf>

Literature Review

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
1	Dual-Mode User Interfaces for Web- Based Interactive 3D Virtual Environme nts Using Three.js	Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo	2017		The dual-mode user interface, which combines 2D and 3D elements, results in quicker retrieval of information compared to using 3D websites alone. Users reported higher satisfaction levels when using the dual-mode interface. The 2D interface achieved the highest rating for user interface quality, while the 3D interface was considered the most aesthetically pleasing.	Presentation and readability of text in the 3D interface were worse compared to the dual-mode and 2D interfaces. The dual-mode interface lacked a content search function, which could potentially slow down data retrieval for users accustomed to using Ctrl+F for searching on web pages.	The research shows promise for dual-mode user interfaces, combining 2D and 3D elements, to enhance user interaction in 3D virtual environments. Future work is suggested to integrate 3D and 2D views more closely and expand the scope of the study to fully assess the merits of dual-mode interfaces. The paper also references related work on integrating 3D objects and 2D HTML elements within 3D space and discusses previous research on dual-mode user interfaces for web content.
2	React JS – An Emerging Frontend Javascript Library	Pratik Sharad Maratkar and Pratibha Adkar	2021		React JS is an emerging and popular frontend JavaScript library known for its fully component-based architecture. It simplifies the development of rich UIs by using reusable	React primarily deals with the View part in MVC, so other tools are needed for backend development. Some developers may find JSX programming challenging during the learning phase. React's environment evolves rapidly,	The paper provides a comparison between React JS and Angular JS, highlighting their differences in terms of development, performance, and usage. It discusses React JS's architecture, including React Virtual DOM, one-way data flow,

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					components. Facebook, Instagram, and other organizations back React JS and use it in their web applications. React JS offers tools for easy debugging, including Chrome extensions.	requiring developers to stay updated with its changes.	React components, and JSX syntax.
3	Frontend Development with React.js	Anjali Rananavare	2022		React.js is widely used in web development, with 46.4% of all websites surveyed using it. React.js simplifies the development of dynamic web applications by requiring less code and offering more functionality. It is an open-source library with a large development community and is used by companies like Facebook, Instagram, and Netflix. React's Virtual DOM and one-way data flow contribute to efficient and faster web application development.	The paper mentions some disadvantages of React.js, such as the need to import libraries for state and model management and the fact that React moves away from class-based modules, which may pose challenges for developers who are more comfortable with Object Oriented Programming.	The paper provides a detailed overview of React.js, its advantages, architecture, and how it compares to other frameworks like Angular. It emphasizes React's role in simplifying frontend development and highlights its popularity in the industry.
4	React Apps with Server-Side Rendering: Next.js	Harish A Jartarghar, Girish Rao Salanke, Ashok Kumar A.R, Sharvani G.S, Shivakumar Dalali	2022		The paper explains that React.js is a flexible library for building reusable User Interface (UI) components, but it primarily relies on client-side rendering, leading to slow page loading. Next.js is introduced as a solution to this problem, utilizing	The paper doesn't explicitly mention any gaps or disadvantages in its research.	The paper provides a brief overview of web development trends, such as the adoption of HTML5 and the development of JavaScript engines like Chrome V8. It discusses the evolution of JavaScript from interacting with CSS to being used for server-side applications with technologies like Node.js. The paper also

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					server-side rendering for improved page loading speed and performance. Next.js helps enhance SEO by allowing search engines to crawl the site more effectively.		introduces the concept of the Document Object Model (DOM) and its significance in web development. Additionally, it explains the benefits of React.js as a component-based library and introduces the concept of the Virtual DOM.
5	React JS (Open Source JavaScript Library)	Alok Kumar Srivastava, Vaishnavi Laxmi, Payal Singh, Km Pratima, Vibha Kirti	2022		The paper discusses the benefits of using React JS for front-end web application development. It emphasizes React's ability to create large and complex web applications that can update data without requiring page refreshes. React JS is praised for providing a better user experience and enabling the development of fast and robust web applications. The paper also mentions that React JS can integrate with other JavaScript libraries or frameworks, including AngularJS.	The paper does not explicitly mention any disadvantages or gaps in React JS but focuses on its advantages and features.	The paper provides an overview of React JS, its history, key features, and its role in modern web development. It discusses the React component lifecycle and its methods, emphasizing the importance of understanding these methods for effective development. The paper also mentions the popularity of React JS and its dominance in the front-end development market
6	Modern Web-Development using React.js	Bhupati Venkat Sai Indla and Yogeshch	2018		ReactJS offers lightweight DOM for better performance, using a virtual DOM to optimize updates to the browser DOM. It has an easy learning curve and uses JSX for simplifying development. ReactJS is known for its high-performance due to the virtual DOM	The paper doesn't delve into specific gaps or disadvantages of ReactJS but mentions some limitations: React primarily handles the View entity in the MVC pattern, requiring additional tools for complete project development. The use of inline templates and JSX may be complex for some developers. Failures in ReactJS	ReactJS provides a modern and efficient approach to web development, especially for building dynamic and interactive user interfaces. It simplifies the development process, offers high performance through virtual DOM, and encourages unidirectional data flow. The paper emphasizes ReactJS's potential to impact the way web

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					<p>feature. Unidirectional data flow is supported, which keeps components synchronized with the data flow. ReactJS uses a virtual DOM to enhance the performance of applications with frequent data changes.</p>	<p>occur at compile time, which can be frustrating for developers compared to runtime errors in other frameworks.</p>	<p>applications are developed and its ability to meet the demands of modern web development trends.</p>
7	Review on React JS	Dimpy Bansal	2020		<p>React JS simplifies the creation of interactive user interfaces by allowing developers to design simple views for each state in their applications.</p> <p>React's component-based architecture promotes reusability, making it easy to manage individual components' states and pass data between them.</p> <p>React's Virtual DOM feature improves site performance by selectively updating parts of the actual DOM, reducing computing power and loading times.</p> <p>React is well-suited for developers familiar with JavaScript, offering a smoother learning curve compared to some other frameworks.</p>	<p>While the paper provides a comprehensive overview of React JS, it does not delve into potential disadvantages or limitations of using React. A more balanced assessment that includes drawbacks or challenges could offer a more complete understanding of the technology</p>	<p>The paper discusses the evolution of web development before and after the introduction of JavaScript libraries like React JS. It highlights the advantages of using React, such as component reusability, improved user interactions, and support for various types of web applications. The paper also suggests potential application areas for React JS, including blogs, business websites, forums, eLearning modules, and more.</p>

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					<p>React's rich ecosystem includes ready-made and customizable components, tools, and tutorials, enabling developers to build web apps more efficiently.</p> <p>React can be used to create single-page applications and cross-platform mobile apps using React Native.</p>		
8	Role of Node.js in Modern Web Application Development	Ghansham Jadhav1, Flavia Gonsalves 2	2020		<p>Node.js is a JavaScript runtime environment built on Chrome's V8 JavaScript engine, designed for server-side applications. It focuses on low memory consumption and performance, making it suitable for building scalable and lightweight applications. Node.js allows developers to use JavaScript for both client and server-side scripting. It uses event-driven, non-blocking, and asynchronous approaches, enabling it to handle concurrent requests efficiently.</p>	The paper does not explicitly mention any gaps or disadvantages in the research.	<p>Node.js's internal structure includes V8, an open-source project by Google, and libuv, an abstraction layer for handling I/O operations. Node.js provides a unified API for JavaScript developers to interact with C/C++ code running in the background. Node.js introduces a modular system for managing dependencies and code isolation. NPM (Node Package Manager) simplifies package management and dependency handling in Node.js applications. Node.js is favored by major companies like PayPal, LinkedIn, Yahoo, Netflix, and GoDaddy for its scalability, performance, and efficiency.</p>
9	Comprehensive Analysis of React-Redux Hybrid App Development Framework	Shravan G V, Prof. Anitha Sandeep	2020		The research aims to judge the user experience of web applications developed using the React-Redux framework as	The research paper does not explicitly mention any gaps or disadvantages in the React-Redux framework. However, it is important to note that	The paper provides insights into the architecture of React-Redux and its components, including components, templates, containers, actions,

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					satisfactory. The paper suggests that creating user interfaces with React-Redux is easier compared to other platforms. React-Redux is capable of generating both simple and complex applications for cross-platform use, with a focus on high data fetching without caching.	while React-Redux offers advantages, it may not be the ideal choice for all types of applications or development scenarios. The paper could have discussed any limitations or challenges faced during the implementation of the framework.	action creators, reducers, selectors, and the store. It also highlights the applications of React-Redux, such as handling front-end data, state interpretation for complex applications, and scalability.
10	Survey And Analysis Of Rendering Realtime 3D Object On Cross-Browser & Cross-Platform Using WebGL	Yogiraj Patil, Kirti Wanjale	2020		The paper explores various rendering approaches and techniques for complex 3D objects in real-time using web browsers and WebGL. It emphasizes the importance of optimized rendering for web-based 3D applications. The paper mentions the use of glTF (GL Transmission Format) for efficient 3D model representation in WebGL.	The paper discusses various aspects of rendering 3D objects but does not provide a comprehensive analysis of performance or specific applications. It does not delve into the use of specific datasets or examples of real-world applications.	The paper acknowledges the challenges of rendering 3D objects in real-time on various devices and screen sizes. It highlights the role of JavaScript frameworks and WebGL in enabling 3D web applications. The paper mentions the importance of efficient data representation using technologies like glTF.
11	Robust Real-Time Shadows for Dynamic 3D Scenes on the Web	Tim Nicolas Eicke, Yvonne Jung, and Arjan Kuijper	2014		The research focuses on improving the quality of real-time shadows in web-based 3D scenes. The authors propose the use of Variance Shadow Maps and PSSM to achieve this. They provide practical implementation results and demonstrate that these techniques significantly enhance shadow	The paper identifies several limitations and challenges in achieving high-quality shadows in web-based 3D scenes. These include the limited capabilities of WebGL compared to other graphics libraries like OpenGL, issues related to bias in shadow mapping, and aliasing artifacts at shadow edges. The paper acknowledges that some adjustments are needed to address these limitations.	The paper highlights the importance of shadows in enhancing the authenticity of virtual 3D scenes and aiding in the perception of spatial relationships. It also mentions that while WebGL has some limitations, it contributes to the success of web-based 3D technology due to its wide platform compatibility.

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					quality, especially in large scenes.		
12	Movie Data Visualization Based on WebGL	Min Li, Chunfang Li	2021		Node.js uses a single-threaded event loop model, making it capable of handling multiple concurrent client requests efficiently.	The paper mentions that 3D data visualization charts are rarely involved in current data visualization practices, but it does not elaborate on the specific limitations or disadvantages of 3D data visualization compared to 2D visualization.	The use of JavaScript for both client-side and server-side development has streamlined the development process for many web applications.
13	Comprehensive Analysis of React-Redux Development Framework	Shravan G V and Prof. Anitha Sandeep	2020		<p>The research work aims to judge the user experience of web applications created using the React-Redux framework as satisfactory. It examines whether user interface creation is easier compared to other platforms. It differentiates between the generation of simple and complex applications using React-Redux. The React-Redux framework allows for the development of compatible code for both Android and iOS platforms, using a single codebase.</p> <p>Redux centralizes state management, making it easier to manage application states.</p> <p>Redux enables features like undo/redo, state persistence, and provides an excellent debugging experience.</p>	The research paper does not explicitly mention any disadvantages or gaps in the React-Redux framework. However, it is essential to note that while React-Redux has many advantages, it may not be suitable for all types of applications, and developers should consider their specific use cases.	The paper provides an overview of the architecture of React-Redux, highlighting components such as Component, Template, Container, Actions & Action Creators, Reducer, Selector, and Store. It discusses the need for Redux in managing application state, especially in complex applications.

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
14	Server- Based Rendering of Large 3D Scenes for Mobile Devices Using G- Buffer Cube Maps	Juergen Doellne, Benjamin Hagedorn	2020		The paper demonstrates that this server-based rendering approach is effective in rendering large 3D scenes on mobile devices. It decouples the complexity of the 3D scene from data transmission complexity, allows for advanced 3D rendering on the server, and provides a high degree of protection for 3D content while supporting interactive user experiences on clients.	The paper does not explicitly mention any disadvantages or gaps in the research.	Different types of 3D visualizations are implemented for film-related data, such as histograms, pie charts, maps, and force-directed graphs.
15	The Research and Design Of 3D Web Guide System Based On WebGL	Cui Peng	2021		The research paper describes the development of a 3D Web guide system that allows users to navigate unfamiliar environments using their mobile phones. It employs WebGL, three.js, and various web technologies to create interactive 3D scenes and calculate optimal paths between scenes. The system is designed for use in large amusement parks and shopping malls, serving as a form of advertising and a new type of 3D web application.	The paper does not explicitly mention any disadvantages or gaps in the research.	The paper outlines the architecture and operation of the 3D Web guide system, which consists of a client-side application running on mobile phones and a server-side system for path calculation and database interaction. The system's performance is described in terms of frame rate and resource usage on both the client and server sides.
16	Performance Optimization using MERN stack on Web Application	Sourabh Mahadev Malewade , Archana Ekbot	2021		The use of React.js, MongoDB, Node.js, and Express.js in building the web application.	The paper doesn't explicitly mention any gaps or disadvantages in the research.	The paper provides a comprehensive overview of the technologies used in building an e-commerce

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					<p>The advantages of using Node.js for asynchronous, event-driven programming.</p> <p>The role of Express.js in simplifying back-end code and providing middleware support.</p> <p>The benefits of React.js in building user interfaces with components.</p>		<p>web application, emphasizing the importance of understanding client demands and the potential for online businesses. It discusses the advantages of the MERN stack and highlights key aspects of each technology used in the project</p>
17	Efficient visualization of 3D models by web browser	Bartosz Sawicki and Bartosz Chaber	2013		<p>The findings of the paper include the successful development of a JavaScript-based web component for 3D model visualization in web browsers. This component adapts to the device's capabilities and provides a natural 3D experience. Usability tests were conducted on real-life users, and the results were positive.</p>	<p>The paper mentions that one disadvantage of their approach is the additional computational effort required on the server side to prepare the mesh before displaying it. It does not go into detail about potential limitations or drawbacks of their method.</p>	<p>The paper discusses the importance of efficient 3D model visualization, particularly for mobile devices with limited computational power and network bandwidth. It introduces the concept of progressive mesh streaming as a way to optimize the transmission of 3D models to clients. The authors also highlight the accessibility of their web-based solution, as web browsers are present on virtually every device.</p>
18	3D Rubik's Cube - Online 3D Modeling System Based on WebGL	Buyun Sheng, Feiyu Zhao, Chenglei Zhang, Xiyan Yin, Yao Shu	2017		<p>The paper demonstrates the development of an online 3D modeling system that allows cloud-based 3D model design. It leverages WebGL for 3D rendering and provides a range of 3D modeling functions. The system was tested for stability and performance, showing good results in terms of frames per second (FPS) and load times compared to</p>	<p>The paper mentions that the system is still in the research and development stage, implying that it may not have a fully matured set of features. It also doesn't discuss any specific disadvantages or limitations of the system.</p>	<p>The paper focuses on the development of an online 3D modeling system suitable for cloud-based 3D printing. It highlights the use of WebGL and Three.js for rendering and provides insights into the improved Phong reflection model and CSG tree-based modeling. The system appears to offer good performance in terms of FPS and load times.</p>

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					another online 3D modeling system called Clara.io.		
19	WEBAPP SERVICE FOR BOOKING	Saundariya K, Prabakaran D,	2021		The research presents a user-friendly website that allows users to easily book handyman services online. It offers various services like cleaning, COVID-sanitization, furniture maintenance, electrical works, appliance repair, house painting, and plumbing. Users can select services, view available professionals based on location and cost, and book them. Handyman professionals can showcase their skills and accept or decline tasks. The system aims to provide a convenient and cost-effective solution for connecting users with professional workers.	The paper does not explicitly mention any gaps or disadvantages in the research.	The research focuses on addressing the increasing demand for handyman services by providing an online platform. It emphasizes the importance of verifying the professionalism of workers through admin approval and offers a user-friendly interface for booking and tracking services.
20	HANDYMAN USING MONGO DB, EXPRESS JS, REACT JS, NODE JS	Abirami M, Srimathi B, Senthil Kumaran R, Nagarajan G (IEEE Member)	2021		The research presents a user-friendly website that allows users to easily book handyman services online. It offers various services like cleaning, COVID-sanitization, furniture maintenance, electrical works, appliance repair, house painting, and plumbing. Users can select services, view	The paper does not explicitly mention any gaps or disadvantages in the research.	The research focuses on addressing the increasing demand for handyman services by providing an online platform. It emphasizes the importance of verifying the professionalism of workers through admin approval and offers a user-friendly interface for booking and tracking services.

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					available professionals based on location and cost, and book them. Handyman professionals can showcase their skills and accept or decline tasks. The system aims to provide a convenient and cost-effective solution for connecting users with professional workers.		
21	Research and Application of Web3D Exhibition Based on WebGL and Html5	M.J. Bian, J. Gao, H.H. Gao, J.P. Xu	2015		<p>The authors have designed a Web3D solution that combines WebGL and HTML5. They developed the Web3D Exhibition Building System (Web3D-EBS) to create Web3D exhibitions in web applications. The solution offers good compatibility and runs without the need for plugins, relying on GPU rendering. The system has been applied to the project of Digital Museums of Colleges and Universities in Shanghai, demonstrating its convenience and effectiveness in Web3D exhibition applications.</p>	<p>The paper does not specify the publication year, making it difficult to determine the currency of the research. It does not delve into the technical details of the Web3D-EBS system, such as implementation specifics or performance benchmarks. The paper does not discuss potential limitations or challenges encountered during the implementation of the solution</p>	<p>The authors note the increasing importance of 3D visualization in web applications for enhancing the user experience. The paper highlights various existing Web3D solutions, such as Java3D, Flash3D, VRML, and Cult3D, and their advantages and disadvantages. WebGL is introduced as a key technology that can efficiently render complex 3D scenes in browsers by leveraging GPU capabilities. HTML5 is mentioned as providing support for 3D graphics in web applications and improving compatibility. Three.js, an open-source JavaScript library built on WebGL, is recommended for enhancing the efficiency and flexibility of building Web3D exhibitions.</p>
22	Web 2.0 and Virtual World Technologies: A Growing	Albert L. Harris and Alan Rea	2009		Web 2.0 and virtual world technologies are becoming increasingly	The paper does not explicitly mention any gaps in research, but it acknowledges the	The paper emphasizes the need for educators to adapt to the changing technological landscape

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
	Impact on IS Education				<p>important in IS education. Students today are tech-savvy and often more knowledgeable about these technologies than their professors. Collaborative learning is being promoted across disciplines, and these technologies facilitate collaborative efforts in education. Different types of Web 2.0 technologies, such as wikis, blogs, podcasts, and social networks, are being used to enhance IS education. Virtual worlds, like Second Life, provide immersive environments for students to experiment and collaborate.</p>	challenges associated with using Web 2.0 and virtual world technologies in education, such as technical requirements, potential disruptions, and issues related to evaluating group work.	and leverage these technologies effectively to engage students and enhance learning. It suggests that instructors should carefully consider how to use these technologies to complement their teaching methods. Additionally, the paper highlights the increasing importance of a global perspective in education and the potential for Web 2.0 technologies to facilitate cross-cultural interactions.
23	Investigating Web3D topics on StackOverflow: a preliminary study of WebGL and Three.js	Farag Almansou ry, Sègla Kpodjedo, and Ghizlane El Boussaidi	2020		<p>WebGL received less community attention compared to Three.js in terms of the number of questions and views. Three.js received significantly more community attention but had lower community support than WebGL. The study identified various tags associated with WebGL and Three.js, including specific technologies (e.g., Pixi.js, A-Frame, Blender, FBX) and development concerns (e.g.,</p>	<p>Limited Scope: The study only considers data from Stack Overflow for the years 2015 to 2019. It may not capture the most recent trends and developments in WebGL and Three.js.</p> <p>Lack of Detailed Analysis: The paper mentions various tags associated with WebGL and Three.js but does not provide an in-depth analysis of each tag's significance or relevance.</p>	The paper highlights the importance of community support and attention for developers working with WebGL and Three.js. It suggests that while Three.js may have a larger community on Stack Overflow, it faces challenges in terms of support compared to WebGL. Developers are advised to consider these findings when making technology choices for their Web3D projects.

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					shader, textures, raycasting, camera). When coupled with Three.js, most technologies experienced a drop in community support, while the reverse was true for WebGL.		
24	A Framework for Browser-based Multiplayer Online Games using WebGL and WebSocket	Bijin Chen, Zhiqi Xu	2011		The framework successfully enables the development of browser-based multiplayer online games with real-time 3D graphics. The performance of the framework was tested using multiple clients, and the measured data aligns with theoretical expectations. The paper provides insights into factors that affect communication efficiency, such as the number of clients, network quality, and hardware capabilities.	The paper mainly focuses on a small group of users, and its scalability to larger multiplayer online games is not discussed. While the paper discusses the impact of various factors on communication efficiency, it doesn't provide detailed optimization strategies.	The use of HTML5, WebGL, and WebSocket technologies makes it easier to create cross-platform browser-based multiplayer online games without the need for explicit installations. The framework's performance is influenced by factors such as the number of clients, network quality, and hardware capabilities. The framework architecture involves separate web and game servers, with web workers optimizing communication. Real-time 3D graphics are made possible through Three.js and real-time communication through jWebSocket.
25	Immersive 3D Modeling with Blender and Off-the-Shelf Hardware	Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo	2020		The paper mentions challenges related to accurate object posing but does not delve into possible solutions or improvements in detail. It does not provide extensive technical details about the implementation of the 3D modeling application. The study sample size was relatively	The paper points out that while the application was effective for some 3D modeling tasks, it had limitations in terms of accurate object positioning. It also mentions the need for further refinement to improve posing accuracy. Additionally, the study focused on short-term use, and long-term usability and	The paper demonstrates the feasibility of creating an immersive 3D modeling application using open-source software and off-the-shelf hardware, making it more accessible to a wider audience of 3D artists. User feedback indicates that the application was enjoyable and had intuitive features, but there were concerns about accuracy and

Sr. No.	Title	Author Name	Year	Published At	Findings	GAP	Future Direction
					small, with 14 participants, and might not represent a broader user base.	comfort were not extensively evaluated.	potential eye fatigue. The study also revealed a mixed perception among participants regarding the future use of immersive technology in 3D modeling, with some believing it would become more common while others were unsure.

2.3. References

- [1] Dual-Mode User Interfaces for Web- Based Interactive 3D Virtual Environments Using Three.js - Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo – 201
- [2] React JS – An Emerging Frontend Javascript Library - Pratik Sharad Maratkar and Pratibha Adkar – 2021
- [3] Frontend Development with React.js - Anjali Ranavare – 2022
- [4] React Apps with Server- Side Rendering: Next.js - Harish AJartarghar, Girish Rao Salanke, Ashok Kumar A.R, Sharvani G.S, Shivakumar Dalali - 2022
- [5] React JS (Open Source JavaScript Library) - Alok Kumar Srivastava, Vaishnavi Laxmi, Payal Singh, Km Pratima, Vibha Kirti - 2022
- [6] Modern Web- Development using React.js - Bhupati Venkat Sai Indla and Yogeshch - 201
- [7] Review on React JS - Dimpy Bansal - 2020
- [8] Role of Node.js in Modern Web Application Development - Ghansham Jadhav¹, Flavia Gonsalves² - 2020
- [9] Comprehensive Analysis of React-Redux Hybrid App Development Framework - Shravan G V, Prof. Anitha Sandeep – 2020
- [10] Survey And Analysis Of Rendering Realtime 3D Object On Cross- Browser & Cross-Platform Using WebGL - Yogiraj Patil, Kirti Wanjale – 2020
- [11] Robust Real-Time Shadows for Dynamic 3D Scenes on the Web - Tim Nicolas Eicke, Yvonne Jung, and Arjan Kuijper - 2014
- [12] Movie Data Visualization Based on WebGL - Min Li, Chunfang Li – 2021

- [13] Comprehensive Analysis of React-Redux Development Framework - Shravan G V and Prof. Anitha Sandeep – 2020
- [14] Server- Based Rendering of Large 3D Scenes for Mobile Devices Using G- Buffer Cube - Mapsn Juergen Doellne, Benjamin Hagedorn – 2020
- [15] The Research and Design Of 3D Web Guide System Based On WebGL - Cui Peng – 2021
- [16] Performance Optimization using MERN stack on Web Application - Sourabh Mahadev Malewade, Archana Ekbot – 2021
- [17] Efficient visualization of 3D models by web browser - Bartosz Sawicki and Bartosz Chaber – 2013
- [18] 3D Rubik's Cube - Online 3D Modeling System Based on WebGL - Buyun Sheng, Feiyu Zhao, Chenglei Zhang, Xiyan Yin, Yao Shu - 2017
- [19] WEBAPP SERVICE FOR BOOKING - Saundariya K, Prabakaran D, - 2021
- [20] HANDYMAN USING MONGO DB, EXPRESS JS, REACT JS, NODE JS - Abirami M,
- [21] Srimathi B, Senthil Kumaran R, Nagarajan G (IEEE Member) - 2021
- [22] Research and Application of Web3D Exhibition Based on WebGL and Html5 - M.J. Bian, J. Gao, H.H. Gao, J.P. Xu - 2015
- [23] Web 2.0 and Virtual World Technologies: A Growing Impact on IS Education - Albert L. Harris and Alan Rea - 2009
- [24] Investigating Web3D topics on StackOverflow: a preliminary study of WebGL and Three.js - Farag Almansoury, Sègla Kpodjedo, and Ghizlane El Boussaidi – 2020
- [25] Immersive 3D Modeling with Blender and Off-the-Shelf Hardware Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo - 2020
- [26] Prasetyo, F. A. (2019). Badan Kepegawaian Negara (BKN). [tribunnewswiki. https://www.tribunnewswiki.com/2019/10/24/badankepegawaian-negara-bkn](https://www.tribunnewswiki.com/2019/10/24/badankepegawaian-negara-bkn) (accessed Nov. 02, 2021).
- [27] Yogyakarta, K. R. I. B. (2021). Wakil Kepala Bkn: Siasn Solusi Benahi Kualitas Data Kepegawaian. Yogyakarta.Bkn.Go.Id. <https://yogyakarta.bkn.go.id/berita/2021/10/wakil-kepala-bkn-siasn-solusi-benahi-kualitas-data-kepegawaian> (accessed Nov. 02, 2021).
- [28] Falih, F. (2018). A Review Study of Information Systems. International Journal of Computer Applications, 179(18), 15–19. doi: 10.5120/ijca2018916307.

- [29] Venkat, B., Indla, S., Puranik, Y., Student, P. G., & College, P. E. S. M. (2021). Review on React JS. Journal Title, 5(4), 1137–1139.
- [30] Bhalla, A., Garg, S., & Singh, P. (2020). Present Day Web-Development Using ReactJS. International Research Journal of Engineering and Technology, 7(5), 1154–1157.

2.4. Conclusion

In conclusion, the "3D Outfit Customizer" project aims to transform the online fashion shopping experience by providing a 3D virtual platform for users to design and customize their outfits. By addressing the limitations of traditional online shopping, this project has the potential to revolutionize the fashion industry and empower users to express their unique style while enjoying an immersive and interactive shopping experience. Despite potential challenges and limitations, the project's advantages in personalization and creativity make it a promising endeavor for modern fashion enthusiasts.

Chapter 3: Proposed Methodology

Content:

- 3.1. Introduction
- 3.2. System Design
 - 3.2.1. Introduction to System Planning
 - 3.2.2. Software Design Approach
- 3.3. Gantt Chart
- 3.4. Time Line Chart
- 3.5. Cost Estimation
 - 3.5.1. Cost Beneficial Analysis
- 3.6. Feasibility
- 3.7. Conclusion

3.1. Introduction

Systems design is a methodical approach to crafting a system. This can be carried out in a top-down or bottom-up manner, but in both cases, the process is systematic and considers all the interconnected aspects of the system being created. This includes everything from the system's structure, the necessary hardware and software, right down to the management and transformation of data as it traverses the system.

Systems design is closely intertwined with systems analysis, systems engineering, and systems architecture. The origins of the systems design approach can be traced back to the period just before World War II when engineers were grappling with intricate problems related to control and communication. They recognized the need to formalize their work into a structured discipline with established methods, especially given the emergence of new fields like information theory, operations research, and computer science.

System design is the process of defining the various elements of a system, encompassing its architecture, modules, components, the interfaces between these components, and the flow of data within the system. Its primary aim is to meet the specific needs and requirements of a business or organization by engineering a coherent and efficiently functioning system.

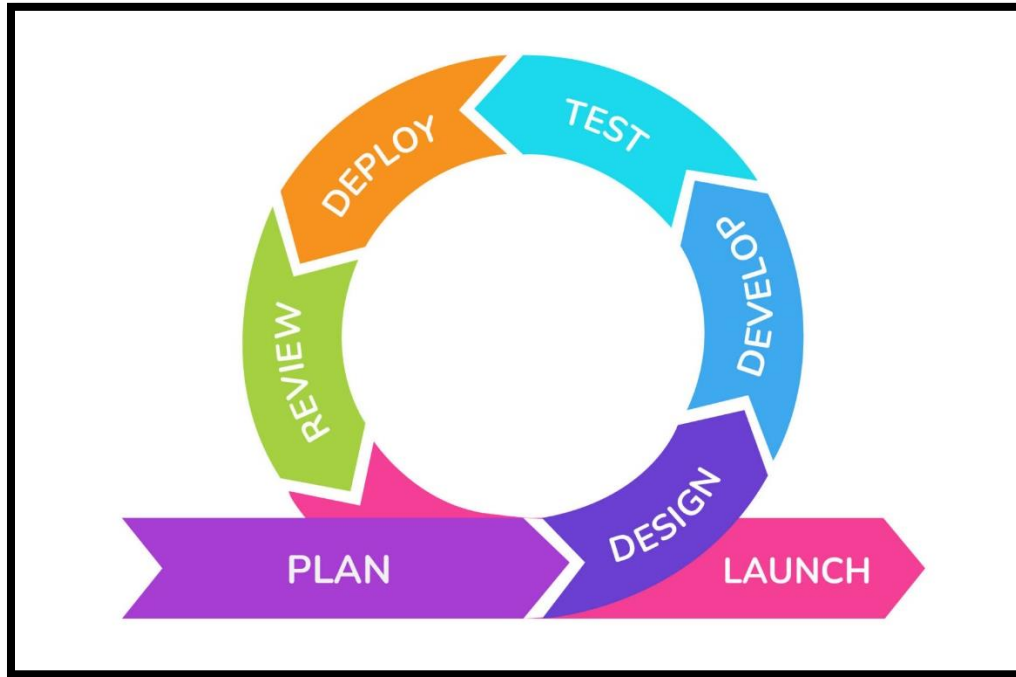
3.2 System Design

3.2.1 Introduction to System Planning

System planning is the process of defining the goals and requirements of a system, determining the resources needed to achieve those goals, and creating a roadmap for implementation. It involves identifying current problems, analyzing them, and designing a solution that meets the needs of the organization or project. System planning typically includes defining objectives, identifying constraints, developing strategies, and creating a timeline for implementation. It is an essential step in the overall system development lifecycle, as it sets the direction for the development and implementation of the system.

3.2.2 Software Design Approach

An Agile process model is a software development methodology that emphasizes flexibility, collaboration, and iterative development. It is characterized by breaking the project into small, manageable increments, allowing for regular reassessment and adaptation based on feedback. Agile methodologies prioritize customer satisfaction, rapid delivery, and the ability to respond to changing requirements.



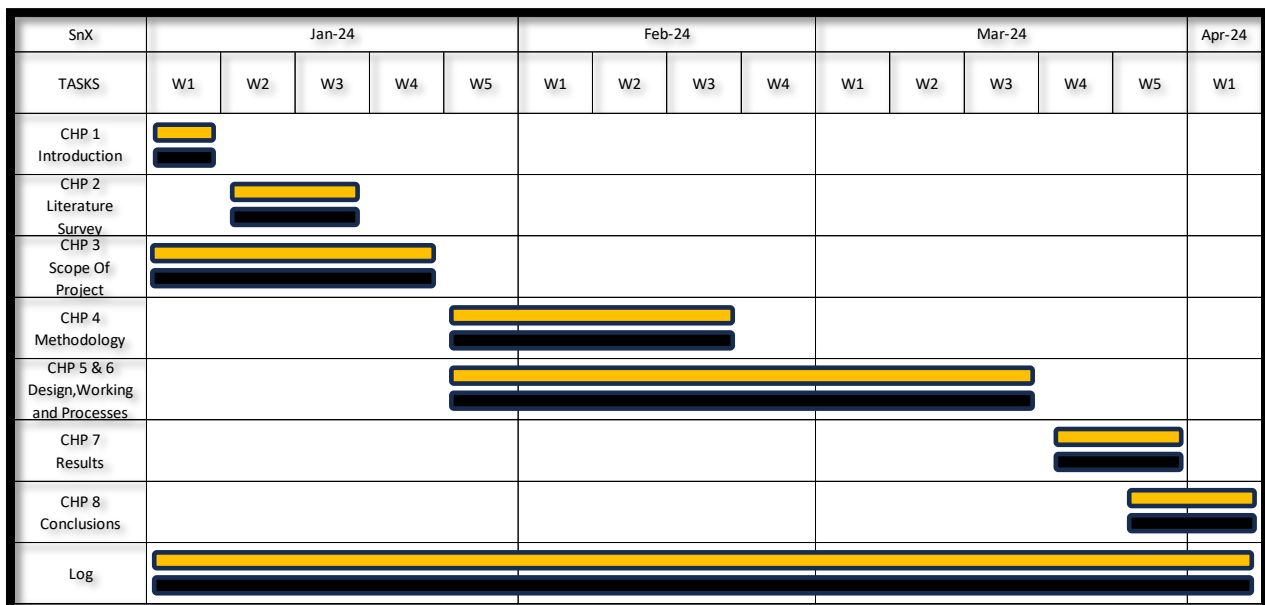
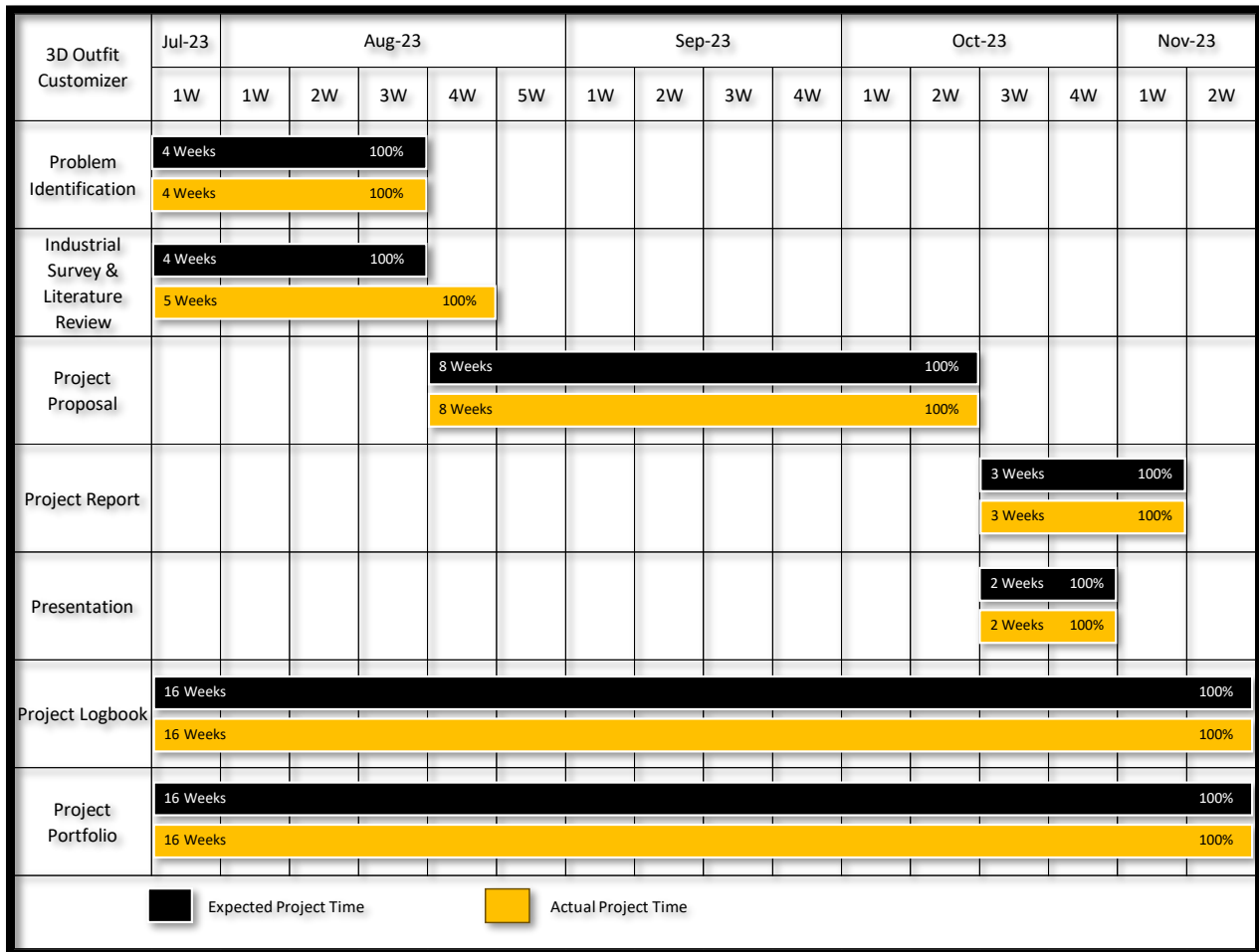
The Agile Software Development process typically consists of the following steps:

1. **Requirements Gathering:** The customer's requirements for the software are gathered and prioritized.
2. **Planning:** The development team creates a plan for delivering the software, including the features that will be delivered in each iteration.
3. **Development:** The development team works to build the software, using frequent and rapid iterations.
4. **Testing:** The software is thoroughly tested to ensure that it meets the customer's requirements and is of high quality.
5. **Deployment:** The software is deployed and put into use.
6. **Maintenance:** The software is maintained to ensure that it continues to meet the customer's needs and expectations.

Principles of Agile Software Development Process are:

1. Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. It welcomes changing requirements, even late in development.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shortest timescale.
4. Build projects around motivated individuals. Give them the environment and the support they need, and trust them to get the job done.
5. Working software is the primary measure of progress.
6. Simplicity the art of maximizing the amount of work not done is essential.

3.3 Time Line Chart



3.4. Gantt Chart

Sr. No.	Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
T1	Problem Identification																
T2	PortFolio P1																
T3	Literature Survey																
T4	Project Proposal																
T5	Action Plan																
T6	PortFolio P2																
T7	Final Report																
T8	Presentation																
T9	Log Book																

TASKS	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
CHP 1 Introduction															
CHP 2 Literature Survey															
CHP 3 Scope Of Project															
CHP 4 Methodology															
CHP 5 & 6 Design,Working and Processes															
CHP 7 Results															
CHP 8 Conclusions															
Log															

3.5 Cost Estimation

In evaluating the "SnX 3D Outfit Customizer" project's economic viability, it's crucial to consider both costs and benefits. Development costs encompass personnel, software tools, and infrastructure expenses, while ongoing operational costs include maintenance, support, and marketing. Additionally, infrastructure and software expenditures, such as hardware, cloud services, and cybersecurity tools, are essential. Benefits may include time savings, improved career outcomes, reduced skills mismatch, increased productivity, and cost savings. Qualitative benefits, like enhanced user satisfaction and societal impacts, also contribute to the project's overall value proposition.

Total Number of Weeks= $16+14=30$ weeks

Total Number of Hours= Total Number of Weeks* Number of hours per week 120

Cost Estimation= Total Number of hours * Cost per hour = $200*$

Total Cost Estimation= Rs. 20, 000

3.5.1. Cost Beneficial Analysis

Cost-benefit analysis (CBA) serves as a crucial method for evaluating the viability of the "SNX 3D Outfit Customizer" project by juxtaposing its costs and benefits. To conduct CBA for this initiative, it's imperative to meticulously identify and quantify both the expenditures and advantages associated with the project. Development costs, inclusive of personnel, software tools, and infrastructure expenses, need thorough consideration, alongside ongoing operational expenses like maintenance, support, and marketing. Furthermore, investments in infrastructure and software, such as hardware, cloud services, and cybersecurity tools, are pivotal. Anticipated benefits range from time savings and improved customization experiences to enhanced user satisfaction and societal impacts. Conducting analyses like net present value (NPV) and return on investment (ROI) enables a comprehensive assessment of the project's financial feasibility, while sensitivity analysis aids in identifying potential risks.

In essence, the evaluation of the "SNX 3D Outfit Customizer" project's economic viability hinges on a rigorous cost-benefit analysis. By meticulously weighing its development and operational costs against the anticipated benefits, including enhanced user experiences and societal impacts, the project's overall value proposition can be determined. Employing financial metrics like NPV and ROI, alongside qualitative assessments, facilitates a comprehensive understanding of the project's feasibility, thereby informing strategic decision-making and ensuring its successful implementation in the fashion industry.

Benefits:

Cost-benefit analysis is a process for weighing the advantages and disadvantages of a project or decision by contrasting the project's expenses and anticipated benefits.

The analysis assists in determining whether the choice or project is financially sustainable and whether the benefits outweigh the expenses. By weighing the costs and benefits of a decision, decision-makers can make more objective and well-informed choices that maximize the value of their resources. Decision-makers may make more informed choices that optimize the value of their resources by using the knowledge about the costs and benefits of various options that cost analysis offers. Understanding the costs involved with a project or choice may help businesses create realistic budgets and more wisely allocate resources.

Cost analysis may be used to find areas where expenses can be cut, including by simplifying operations or negotiating better prices with suppliers. Measures project success: Organizations may evaluate the effectiveness of a project and pinpoint areas for improvement by comparing the actual expenses of a project to its anticipated benefits.

3.6. Feasibility

The feasibility of the "SnX - 3D Outfit Customizer" project involves examining technical, economic, operational, and schedule aspects. Technical feasibility entails evaluating available technology and resources for developing the platform. Economic feasibility involves weighing the costs against the anticipated benefits to determine if the project is financially viable. Operational feasibility assesses the integration of the customizer into existing systems and its acceptance by users. Schedule feasibility ensures that the project can be completed within the allocated time frame. These factors collectively determine the project's likelihood of success and highlight any potential challenges that need to be addressed during planning and execution.

Regarding economic viability, it's crucial to ascertain whether the cost of implementing the 3D outfit customizer is justified by its benefits. Conducting cost-benefit analyses reveals whether the system's benefits outweigh its costs. By comparing the customizer's development costs, including personnel, computing resources, and software expenses, to the potential savings and benefits it offers, we can determine its affordability and practicality. Additionally, considering that development costs are typically one-time expenses, it's essential to ensure that the investment yields sufficient returns in the form of enhanced user experiences, increased engagement, and competitive advantage in the fashion industry.

3.4. Conclusion

In conclusion, Agile methodologies are the perfect fit for the "3D Outfit Customizer" project in the fashion industry. Let's break down why Agile is the way to go: First off, Agile's customer-centric approach is spot-on for a project catering to the diverse and ever-evolving tastes of fashion enthusiasts. By involving users in the development process, the "3D Outfit Customizer" can be tailored to meet individual needs and unique styles. The fashion industry's fast-paced nature requires adaptability, and Agile delivers. Its flexibility enables the project team to swiftly respond to shifting trends and requirements, ensuring the platform's continued relevance and competitiveness. Agile's iterative development approach is another big win. It means we can get valuable features into users' hands earlier, allowing them to start creating and customizing outfits sooner, enhancing the overall experience. Open and transparent communication is at the core of Agile. This is vital for understanding and adapting to the ever-changing world of fashion, ensuring the project stays in sync with customer needs. Last but not least, Agile's emphasis on quality assurance is a big deal. It ensures the final product meets high standards, which is a critical factor for success in the fashion industry.

So, in a nutshell, Agile's customer-centric, adaptable, and collaborative approach, coupled with its focus on risk management and quality assurance, makes it the top choice for the "3D Outfit Customizer" project. Going Agile will help us tackle the unique challenges of the fashion industry and deliver a more personalized, competitive, and high-quality solution to our users.

Chapter 4: System Design

Content:

- 4.1 Introduction
- 4.2 Block Diagram
- 4.3 System Architecture
- 4.4 Data Flow Diagram
- 4.5 Table Structure
- 4.6 State Transition Diagram
- 4.7 E-R Diagram
- 4.8 Conclusion

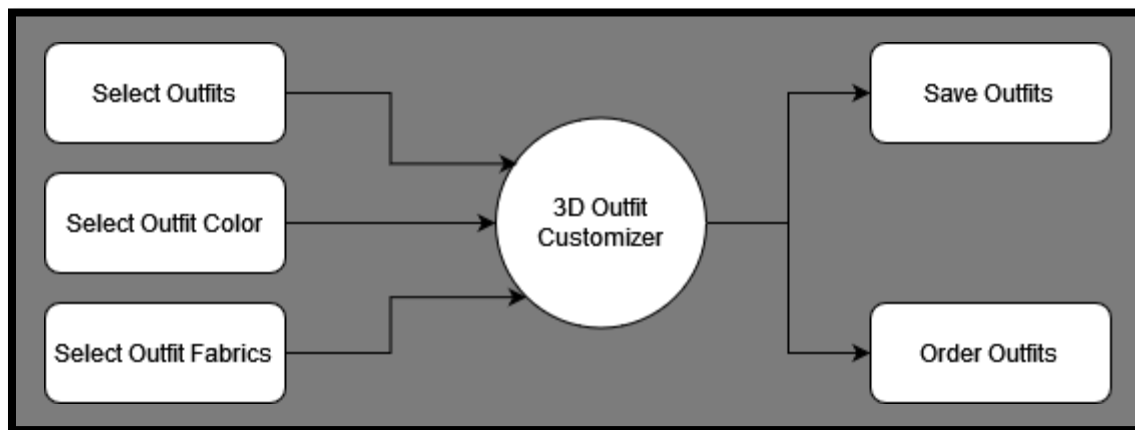
4.1 Introduction:

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system. This innovative project leverages the power of artificial intelligence (AI) and data analysis to streamline the career decision-making process.

In this chapter we will be designing the block diagram, system architecture diagram, data flow diagram and also will plan the software design approach.

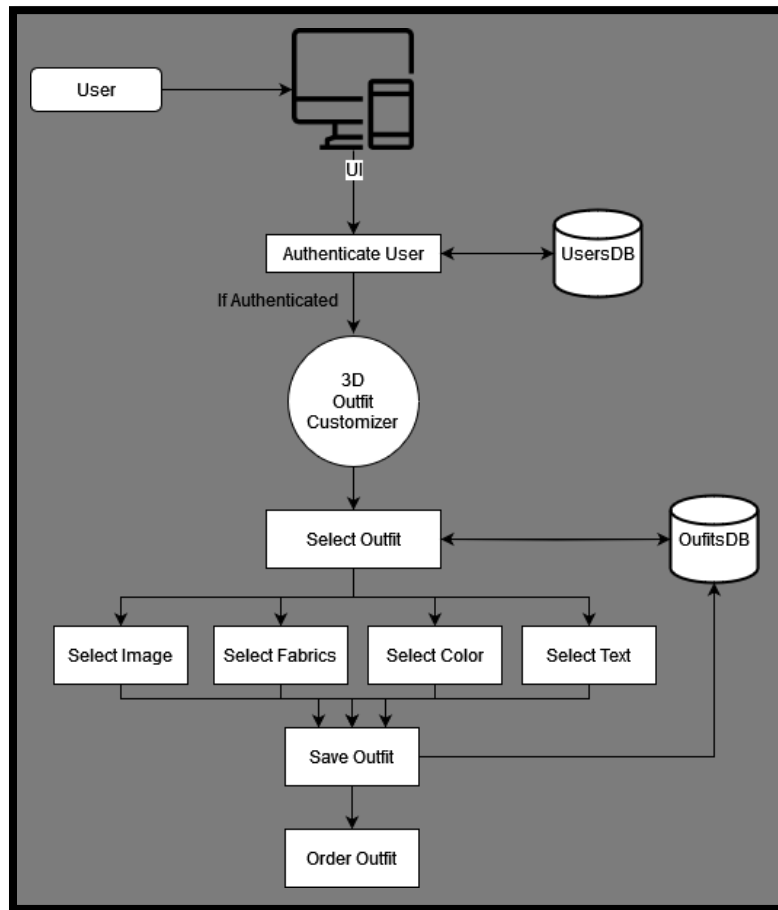
4.2 Block Diagram

A block diagram is a visual representation of a system that uses simple, labeled blocks that represent single or multiple items, entities or concepts, connected by lines to show relationships between them. An entity relationship diagram (ERD), one example of a block diagram, represents an information system by showing the relationships between people, objects, places, concepts or events within that system. Block diagrams are used heavily in engineering and design of diagrams for electronics, hardware, software and processes. Most commonly, they represent concepts and systems in a higher level, less detailed overview. The diagrams are useful for troubleshooting technical issues



4.3 System Architecture Diagram

A system architecture diagram would be used to show the relationship between different components. Usually, they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them. However, it can also be created for web applications. For a web application the system architecture design would include components such as, database, application server, web server, internet, browser etc. Not all of these have to be included in the diagram and there are other components that can be included. Systems in a higher level, less detailed overview. The diagrams are useful for troubleshooting tech



4.4 Data Flow Diagram

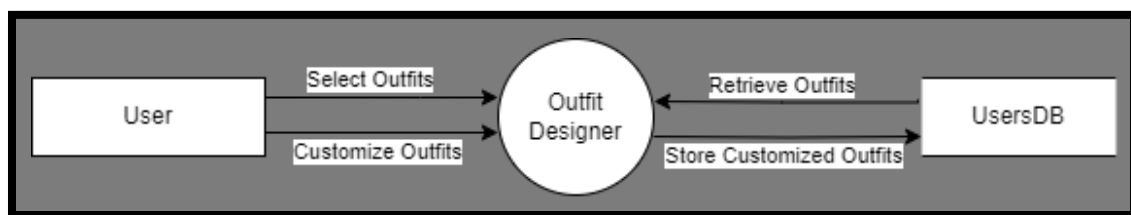


Figure 4.4(a) DFD Level 0 for 3D Outfit Customizer

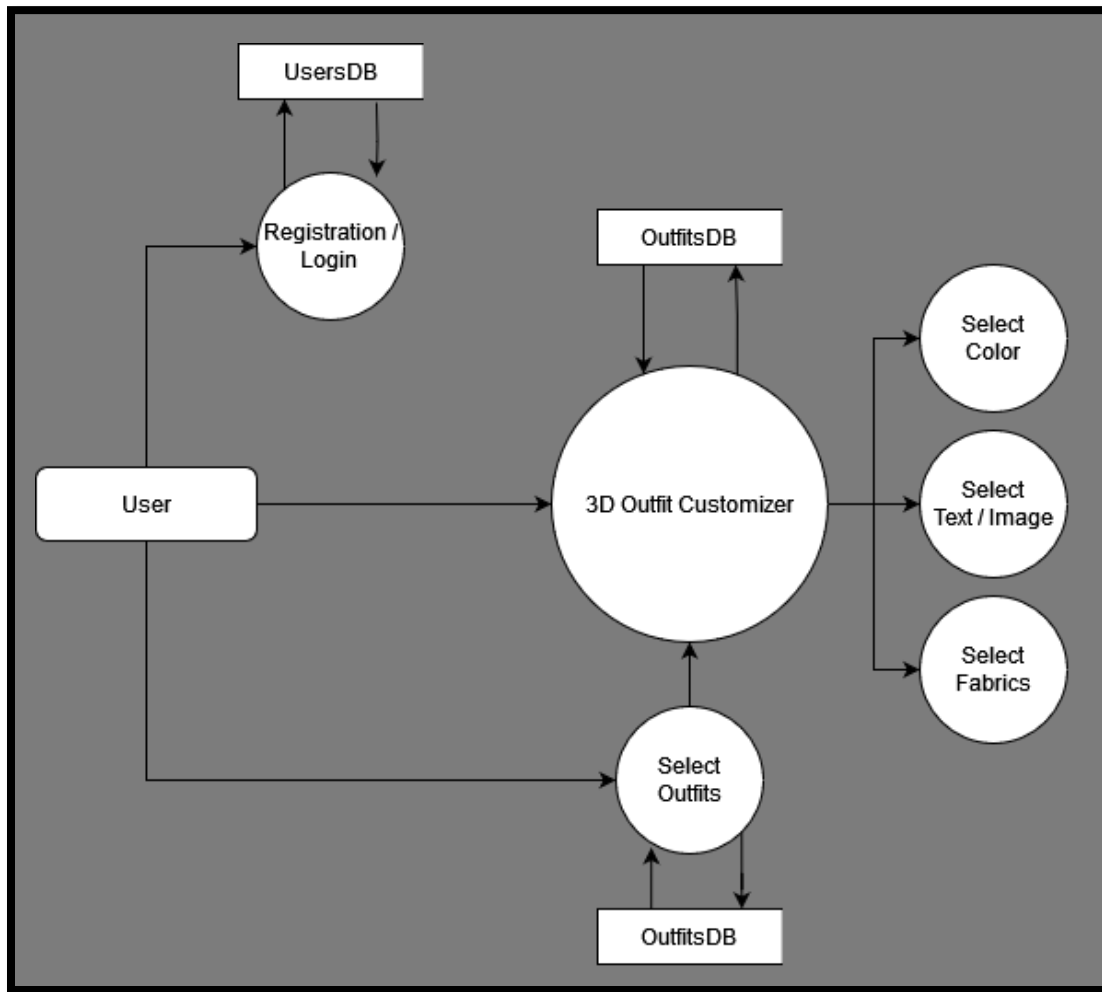
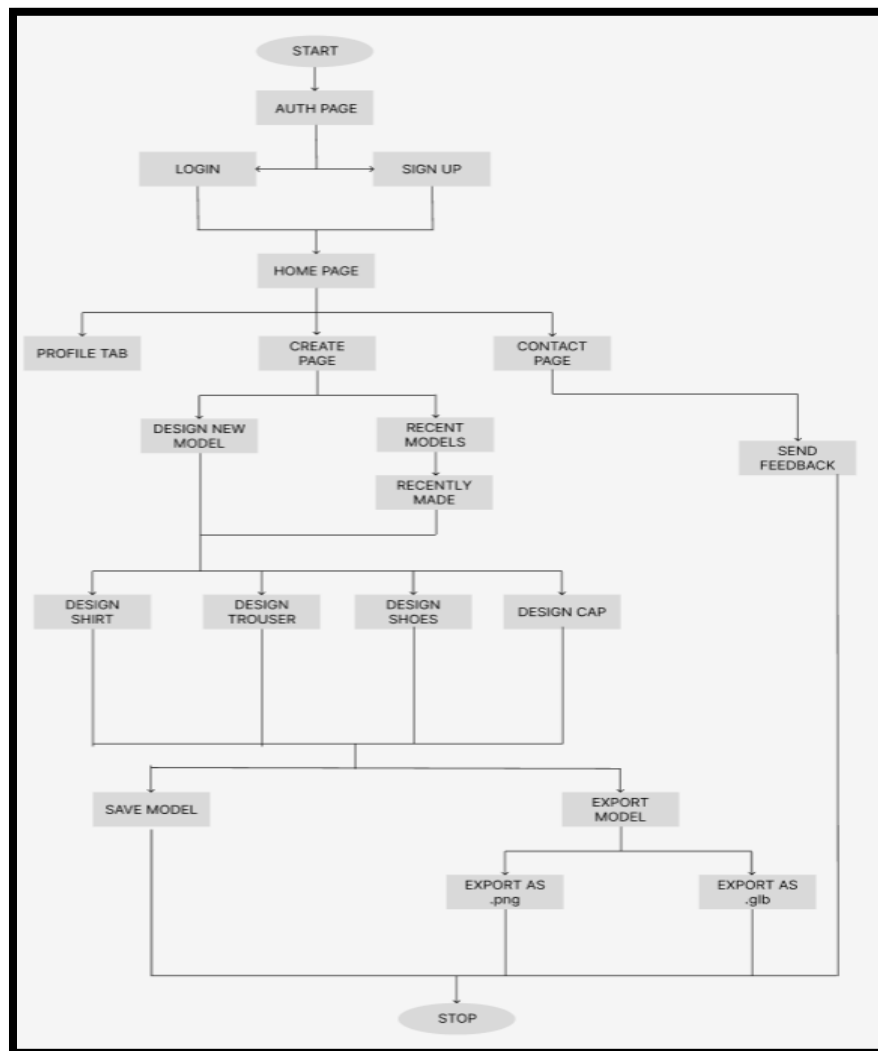


Figure 4.4(b) DFD Level 1 for 3D Outfit Customizer

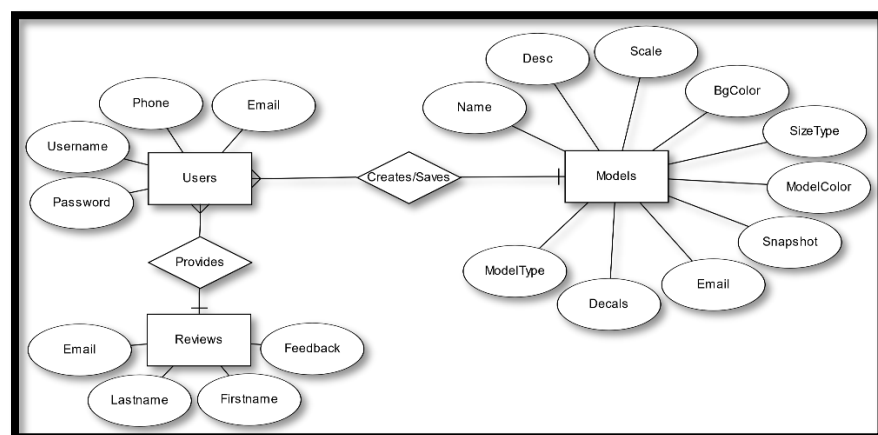
4.5 Table Structure

USERS	MODELS	REVIEWS
Name Email Phone Number Password	Name Description Owner Type Color Text Background Color AI Images Decals Size	First Name Last Name Email Feedback

4.6 State Transition Diagram



4.7 ER Diagram



Chapter 5: Implementation

5.1. Introduction

5.2. Algorithm

5.3. Flowchart

5.4. Coding

5.5. Conclusion

5.1 Introduction

This chapter shows the actual implementation of our project. The algorithm shows the stepwise direction of the project. Then we have implemented the flowchart with the help of algorithm and lastly the code of our project is shown. Seeing this chapter a person will be able to understand our project completely.

5.2 Algorithm

Step 1 : Login or Signup: Access your SnX account by logging in or signing up.

Step 2 : Navigate to Options: Choose from Create Page, Contact Page, or Profile Tab.

Step 3 : Contact Page:

- Send Feedback: Provide feedback through the Contact Page.

Step 4 : Profile Tab:

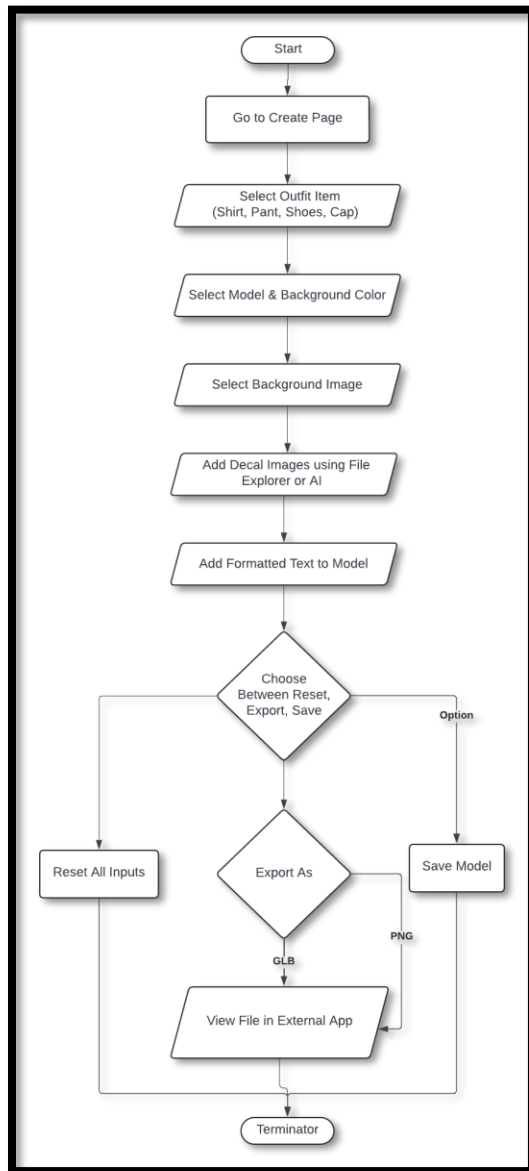
- View Profile Details: Navigate to the Profile Tab to view your profile information.

Step 5 : Create Page:

- Design a New Model: Create a new outfit ensemble comprising Shirt, Trouser, Shoes, and Cap.
 - Customize Model:
 - Choose Model Color, Background Color, and Size.
 - Add AI Images, Decals, and Texts.
 - Save the Model:
 - Assign a Model Name and Description.
 - Save the model to the database.
 - Reset the Model: Reset all customization settings.
 - Export the Model:
 - Export as .png or .glb format.
- Design a Recent Model:
 - Choose a recent outfit ensemble.
 - Customize Model:
 - Choose Model Color, Background Color, and Size.
 - Add AI Images, Decals, and Texts.
 - Save the Model:
 - Assign a Model Name and Description.
 - Save the model to the database.
 - Reset the Model: Reset all customization settings.

- Export the Model:
 - Export as .png or .glb format.

5.3 Flowchart



5.4 Coding

/page.js

```

"use client";
export default function Home() {
  return (
    <
      <motion.div variants={scaleUp} initial="initial" animate="show" style={{paddingLeft: "2rem"}}>
    >
  )
}

```

```

        <StyledProgressBar/>
        <HomeTitle/>
        <HomeWorks/>
        <HomeCategories/>
        <HomeReviews/>
      </motion.div>
    </>
  );
}

```

/layout.js

```

import { Inter } from 'next/font/google'
import './globals.css'
import React from "react";
import IsNavFooter from "@components/IsNavFooter";

```

```
const inter = Inter({ subsets: ['latin'] })
```

```

export default function RootLayout({ children }) {
  return (
    <html lang="en" className="dark">
      <body className={inter.className}>
        <div className="container">
          <IsNavFooter>
            {children}
          </IsNavFooter>
        </div>
      </body>
    </html>
  )
}

```

```

export const metadata = {
  title: 'SnX',
  description: '3D Outfit Customizer',
  manifest: '/manifest.json'
}

```

/create/page.jsx

```

"use client"
export default function Page() {
  const router = useRouter()
  const [models, setModels] = useState([])
  const [loading, setLoading] = useState(true)
  const arr = [0, 1, 2, 3]
  const {user, setSaved} = useAppStore()
  const {reset} = useStore()
  const {setDecalsData} = useDecalStore()

  const fetchModels = async () => {
    const res = await fetch('/api/get/recent-models?user=' + localStorage.getItem('user'))
    const data = await res.json()
    if (data.models)
      setModels(data.models)
    console.log("DATA.MODELS", typeof data.models, data.models)
  }
}

```

```

useEffect(() => {
  reset()
  setSaved({state: true, isUnSaved: true, name: "", desc: ""})
  setDecalsData([])
  fetchModels().then(() => setLoading(false))
}, [user]);

const openModel = (model) => {
  if (!model) return
  router.push(`/create/recent/${model.modelType}/${model._id}`)
}

return (
  <CreatePageContainer>
    <CardGrid>
      <CardGridHeader>
        NEW
      </CardGridHeader>
      {
        CreateOptions.map((item, index) => (
          <StyledBlock
            key={index}
            id={item.id}
          >
            <BlockIcon
              href={item.path}
            >
              {item.icon}
            </BlockIcon>
          </StyledBlock>
        ))
      }
    </CardGrid>
    <Spacer x={20}/>
    <CardGrid
      id="recents"
    >
      <CardGridHeader>
        RECENT
      </CardGridHeader>
      {
        arr.map((idx) => (
          <StyledBlock
            key={idx}
            onClick={() => openModel(models[idx])}
          >
            <img
              className={!models[idx] ? "inactive" : ""}
              src={models[idx]?snapshot || '/images/cube.svg'}
              alt={models[idx]?name}
            />
          </StyledBlock>
        ))
      }
    </CardGrid>
  )
)

```



```

        </CardGrid>
      </CreatePageContainer>
    );
  }

```

/create/new/shirt/page.jsx

```

const Page = () => {
  return (
    <>
      <Studio mode={"new"} modelUrl={'/models/shirt/tshirt.glb'} />
    </>
  )
}
export default Page

```

/components/Studio.jsx

```

export default function Studio({mode, modelUrl}) {
  return (
    <>
      <div className={styles.wrapper}>
        <div className={styles.main} id="studio-viewer">
          <Viewer mode={mode} modelUrl={modelUrl}/>
        </div>
        <div className={styles.cpWrapper} id="design-options">
          <ControlPanel/>
        </div>
      </div>
    </>
  )
}

```

/components/three/Viewer.js

```

const Viewer = ({mode, DD, modelUrl}) => {
  const [modelRayData, setModelRayData] = useState(null)
  const [azimuthPose, setAzimuthPose] = useState(0)
  const [polarPose, setPolarPose] = useState(0)
  const [dpr, setDpr] = useState(2)
  const groupRef = useRef(null)
  const modelRef = useRef(null)
  const {user} = useAppStore()
  const pathname = usePathname()
  const isShirt = pathname.includes('/shirt')
  const isPant = pathname.includes('/pant')
  const isShoe = pathname.includes('/shoes')
  const isCap = pathname.includes('/cap')

  const store = useStore();

  const {decalsData, setDecalsData} = useDecalStore()
  const {saved, setSaved} = useAppStore()

  useEffect(() => {
    if (modelRef.current === null) return;
    if (modelRef.current) {
      console.log('Model reference is now available')
    }
  }, [modelRef])
}

```

```

    }
    decalsData.map((dec) => {
      new THREE.TextureLoader().load(dec.texture, (decalTexture) => {
        const { x: x1, y: y1, z: z1 } = dec.position
        const { x: x2, y: y2, z: z2 } = dec.normal

        const { decal, key } = createDecal(
          modelRef.current, // Geometry
          new THREE.Vector3(x1, y1, z1), // Position
          new THREE.Vector3(x2, y2, z2), // Normal
          decalTexture, // Texture
          dec.size // Size of longest side
        )

        // Add decal to decal manager
        addDecalImages({ path: dec.texture, name: dec.textureName, key: key })

        // Add decal to state
        addDecal({ mesh: decal, key: key })
      })
    })
    // setDecalsData([])
  }, [modelRef.current])

// KEYDOWN
useEffect(() => {
  function handleKeyDownEvent(e) {
    const { key } = e
    if (!animation && !decalPath && !isWriting) {
      if (key === "ArrowLeft") {
        setAzimuthPose((prev) => (prev - 90))
      }
      if (key === "ArrowRight") {
        setAzimuthPose((prev) => (prev + 90))
      }
      if (key === "ArrowUp") {
        setPolarPose((prev) => (prev - 90))
      }
      if (key === "ArrowDown") {
        setPolarPose((prev) => (prev + 90))
      }
      key === "r" && setAzimuthPose((prev) => (prev + 90)) //setModelFlipped((prev) => (prev ? false : true))
    }
    if (decalPath && !isWriting) {
      e.preventDefault() // prevent arrow scrolling
      key === "ArrowUp" && incrementDecalSize(0.01)
      key === "ArrowDown" && decrementDecalSize(0.01)
    }
  }

  function handleMouseWheelEvent(event) {
    if (!decalPath) return

    if (event.deltaY < 0) {
      incrementDecalSize(0.01)
    } else if (event.deltaY > 0) {
      decrementDecalSize(0.01)
    }
  }

```

```

    }

    document.addEventListener('wheel', handleMouseWheelEvent);
    document.addEventListener("keydown", handleKeyDownEvent)
    return () => {
        document.removeEventListener("keydown", handleKeyDownEvent)
        document.removeEventListener('wheel', handleMouseWheelEvent);
    }
}, [isWriting, decalPath, animation, azimuthPose]) //eslint-disable-line

// FLIP ANIMATION
const flipModelAnimation = useSpring({
    config: { tension: 300, mass: 1.3 },
    rotation: [THREE.MathUtils.degToRad(polarPose), THREE.MathUtils.degToRad(azimuthPose), 0],
    onChange: () => invalidate(),
})

const exporter = new GLTFExporter();
const link = useRef();
const toastRef = useRef();

function save(blob, filename) {
    link.current.href = URL.createObjectURL(blob);
    link.current.download = filename;
    link.current.click();
}

const exportScene = () => {
    exporter.parse(groupRef.current, (result) => {
        if (result instanceof ArrayBuffer) {
            save(new Blob([result], { type: 'application/octet-stream' }), 'model.glb');
        } else {
            const output = JSON.stringify(result, null, 2);
        }
    }, (err) => {
        console.error(err.message)
    }, {
        binary: true, onlyVisible: true
    });
};

const items = [
    {
        label: 'Export as Png', id: "export-pic-button", icon: 'pi pi-image', command: () => {
            const imgData = gl.domElement.toDataURL("image/png")
            const a = document.createElement("a");

            a.setAttribute("download", "")
            a.setAttribute("href", imgData)
            a.click()

            showToast("success", "Exported as Png", "Scene has been exported as png", toastRef)
        }
    }, {
        label: 'Export as Glb', id: "export-model-button", icon: 'pi pi-box', command: () => {
            exportScene()
            showToast("success", "Exported as GLB", "Scene has been exported as glb", toastRef)
        }
    }
]

```

```

    },];

const handleReset = (event) => {
  confirmPopup({
    target: event.currentTarget,
    message: 'Are you sure you want to proceed?',
    icon: 'pi pi-exclamation-triangle',
    defaultFocus: 'reject',
    acceptClassName: 'p-button-danger',
    acceptIcon: 'pi pi-check',
    rejectIcon: 'pi pi-times',
    rejectClass: 'p-button-sm',
    acceptClass: 'p-button-outlined p-button-sm',
    accept: () => {
      showToast("info", "Scene Reset", "Scene has been reset", toastRef)
      reset()
    },
    reject: () => console.log()
  });
}

const handleSaveBtnClick = () => {
  if (saved.state) return;
  if (saved.isUnSaved) {
    setDialogVisible(true);
    return;
  }
  handleSave().then()
}

const handleSave = async () => {
  setIsWriting(true)
  if (!saved.name) {
    showToast("error", "Save Failed", "Model name and description are required", toastRef)
    setSaved({...saved, state: false, isUnSaved: true})
    setIsWriting(false)
    return;
  }
  if (saved.isUnSaved) {
    const res = await fetch('/api/post/save-model/verify', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        name: saved.name,
        username: saved.name
      })
    })

    const data = await res.json()

    if (data.exists) {
      showToast("error", "Save Failed", "Model name already taken", toastRef)
      setSaved({...saved, state: false, isUnSaved: true})
      return;
    }
  }
}

```

```

const imgData = gl.domElement.toDataURL("image/png")
const snapshot = await process_image(imgData)
const pathParts = pathname.split('/');
const modelType = pathParts[3];
const savedItems = {
  backgroundColor,
  decalsData,
  modelColor,
  email: localStorage.getItem("user"),
  name: saved.name,
  desc: saved.desc,
  sizeType,
  scale,
  snapshot,
  modelType
}
setSaved({...saved, state: true, isUnSaved: false})
setLoading(true)
setDialogVisible(false);

const res = await fetch('/api/post/save-model', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({savedObj: savedItems})
})

const data = await res.json()
setLoading(false)
setIsWriting(false)
showToast("success", "Saved", "Model has been saved", toastRef)
setSaved({...saved, state: true, isUnSaved: false})
}

useEffect(() => {
  if (saved.isUnSaved) {
    setSaved({...saved, state: false, isUnSaved: true})
  } else {
    setSaved({...saved, state: false, isUnSaved: false})
  }
}, [backgroundColor, backgroundImage, decalImages, decalPath, decalName, decals, gl, modelColor, modelRotation, sceneRef, set, sizeType]);

const [dialogVisible, setDialogVisible] = useState(false);

const footerContent = (<SavePopupFooter>
  <div className="btn-cancel">
    <Button label="Cancel" onClick={() => setDialogVisible(false)}/>
  </div>
  <div className="btn-ok">
    <Button label="Save" onClick={handleSave}/>
  </div>
</SavePopupFooter>);

return (
  <CanvasBackground>

```

```

{loading && (<
  <LoadingOverlay/>
  <SavingScreen>
    <div>
      
    </div>
  </SavingScreen>
</>)}
<Hotkeys/>
<Toast ref={ toastRef}/>
<ConfirmPopup/>
<Dialog visible={ dialogVisible } modal header={ "Save Model" } footer={ footerContent }
  style={ { width: '30rem', zIndex: 10000 } }
  onHide={ () => setDialogVisible(false) }>
  <SavePopup>
    <div>
      <label>
        Model Name:
      </label>
      <input type="text" onChange={ (event) => setSaved({ ...saved, name: event.target.value }) }/>
      <label>
        Model Desc:
      </label>
      <input type="text" onChange={ (event) => setSaved({ ...saved, desc: event.target.value }) }/>
    </div>
  </SavePopup>
</Dialog>
<div className="export-glb-div">
  <a href="" ref={ link }></a>
  <Button id="save-button" label={ "Save" + (saved.state ? "" : " *") } icon={ "pi pi-save" }
    onClick={ handleSaveBtnClick } raised/>
</Button>
  <SplitButton id="export-container" label={ "Export" } icon={ "pi pi-file-export" } model={ items }
    onClick={ async () => {
      } } raised/>
  <Button id="reset-button" label={ "Reset" } icon={ "pi pi-eraser" } onClick={ handleReset } raised/>
</div>
<CreateCanvasWrapper
  camera={ { position: [0, 0, 2.2], fov: 50 } }
  dpr={ dpr }
  frameloop="demand"
  gl={ { preserveDrawingBuffer: true } }
  raycaster={ { far: 3.5 } }
  onCreated={ (state) => setGl(state.gl) }
  shadows
  style={ { cursor: decalPath ? "none" : "auto", backgroundColor: "#101010" } }
>
  <DecalHelper modelRayData={ modelRayData } size={ decalSize }/>
  <Suspense fallback={ null }>
    {
      isShirt &&
      <ShirtModel
        modelRef={ modelRef }
        groupRef={ groupRef }
        url={ modelUrl }
        rotation={ flipModelAnimation.rotation }
        setModelRayData={ setModelRayData }
      />
    }
  </Suspense>
</CreateCanvasWrapper>

```

```

    }
    {
      isPant &&
      <PantModel
        modelRef={modelRef}
        groupRef={groupRef}
        url={modelUrl}
        rotation={flipModelAnimation.rotation}
        setModelRayData={setModelRayData}
      />
    }
    {
      isShoe &&
      <ShoeModel
        modelRef={modelRef}
        groupRef={groupRef}
        url={modelUrl}
        rotation={flipModelAnimation.rotation}
        setModelRayData={setModelRayData}
      />
    }
    {
      isCap &&
      <CapModel
        modelRef={modelRef}
        groupRef={groupRef}
        url={modelUrl}
        rotation={flipModelAnimation.rotation}
        setModelRayData={setModelRayData}
      />
    }
  </Suspense>
  <Scenes/>
  </CreateCanvasWrapper>
  <Loader/>
  </CanvasBackground>)
}

```

export default Viewer

/components/three/ShirtModel.js

```

const ShirtModel = ({modelRef, groupRef, url, rotation, setModelRayData}) => {

  // GLOBAL STATE
  const store = useStore()

  const {addDecalData} = useDecalStore()

  // STATE
  const timestamp = useRef(window.performance.now())

  // ANIMATE
  useFrame(() => {
    if (animation === "animation_360") {
      groupRef.current.rotation.y += (Math.PI * 2) / (60 * 7)
    }
  })
}

```

```

    })
    useEffect(() => {
      if (!animation) groupRef.current.rotation.y = 0
    }, [animation])

    // LOAD MODEL
    const gltf = useLoader(GLTFLoader, url)

    // ADD DECAL TO ARRAY
    const handleDecal = (e) => {
      if (decalPath === null) return
      // Get texture
      process_image(decalPath).then(image => {
        new THREE.TextureLoader().load(image, (decalTexture) => {
          const { x: x1, y: y1, z: z1 } = e.intersections[0].point
          const { x: x2, y: y2, z: z2 } = modelRef.current.localToWorld(e.intersections[0].face.normal)

          const { decal, key } = createDecal(modelRef.current, // Geometry
            new THREE.Vector3(x1, y1, z1), // Position
            new THREE.Vector3(x2, y2, z2), // Normal
            decalTexture, // Texture
            decalSize // Size of longest side
          )

          const obj = {
            key: key,
            model: {...modelRef.current},
            position: {x: x1, y: y1, z: z1},
            normal: {x: x2, y: y2, z: z2},
            texture: image,
            textureName: decalName,
            size: decalSize
          }

          addDecalData(obj)

          // Add decal to decal manager
          addDecalImages({path: image, name: decalName, key: key})

          // Add decal to state
          addDecal({mesh: decal, key: key})

          // Remove decal for one time use
          setDecalPath(null)
          setDecalName(null)

          // Reset decal size
          setDecalSize(initialDecalSize)
        })
      })
    }

    // PASS RAYCAST
    const passRaycast = (e) => {
      // Only submit while decal is active
      if (decalPath) {
        if (timestamp.current + 16 <= window.performance.now()) {
          // DEBOUNCE: Update timestamp for new interval

```



```

        timestamp.current = window.performance.now()

        // Get position
        const posV = e.point.clone()

        // Get world normal
        const n = e.face.normal.clone()
        const nWorld = modelRef.current.localToWorld(n)

        // Set pos and normal
        setModelRayData({position: posV, normalWorld: nWorld})
    }
}

// RESET RAYCAST POS AND NORMAL
const removeRaycast = () => setModelRayData(null)

return (<a.group ref={groupRef} rotation={rotation} castShadow scale={scale}>
    <mesh
        ref={modelRef}
        onPointerMove={passRaycast}
        onPointerOut={removeRaycast}
        onPointerDown={handleDecal}
        geometry={gltf.scene.children[0].geometry}
        castShadow
    >
        <meshStandardMaterial
            color={modelColor}
        />
    </mesh>
    <Decals decals={decals}/>
</a.group>)
}

export default ShirtModel

/components/three/Decals.js
const Decals = ({decals}) => {
    return <mesh>{decals.map((decal) => decal.mesh)}</mesh>
}
export default Decals

// INIT
const helper = new THREE.Object3D(), eulerHelper = new THREE.Euler(0, 0, 0), posHelper = new THREE.Euler(0, 0, 0);

export const createDecal = (model, position, normal, activeDecal, maxSize) => {
    // INFO: This component works with a state array in the parent that holds the decal meshes.
    extend({DecalGeometry})

    // Check for active decal
    if (!activeDecal) {
        window.alert("Choose a design to apply to the product.")
        return {decal: null, key: null}
    }

    // Texture settings

```

```

activeDecal.anisotropy = 16

// Get texture size
const dimensions = changeDimensions(activeDecal.image.width, activeDecal.image.height, maxSize)

// ORIENTATION
const n = normal.clone();
n.add(position)

helper.position.copy(position)
helper.lookAt(n)

// COUNTERACT ROTATION
eulerHelper.setFromRotationMatrix(model.matrixWorld)

// COUNTER POSITION
posHelper.setFromVector3(model.localToWorld(model.position))

// KEY
const key = Math.floor(Math.random() * 999999)

// Return decal
return {
  decal: (<mesh
    key={key}
    name="decal"
    rotation={[eulerHelper.x * -1, eulerHelper.y * -1, eulerHelper.z * -1,]}
    position={[posHelper.x * -1, posHelper.y * -1, posHelper.z * -1,]}
  >
    <decalGeometry
      args={[model, position, helper.rotation, new THREE.Vector3(dimensions[0], dimensions[1], 0.3),]}
    />
    <meshStandardMaterial
      side={THREE.DoubleSide}
      shininess={3}
      map={activeDecal}
      transparent={true}
      anisotropy={16}
      polygonOffset={true}
      polygonOffsetUnits={-100}
    />
    </mesh>), key: key,
}
}

```

/components/controlPanel/ControlPanel.js

```

export default function ControlPanel() {
  const customBgRef = useRef()
  const store = useStore()

  useEffect(() => {
    function handleKeyDownEvent(event) {
      const {key} = event
      key === "Escape" && setDecalPath(null)
    }
  })
}

```

```

    document.addEventListener("keydown", handleKeyDownEvent)
    return () => {
        document.removeEventListener("keydown", handleKeyDownEvent)
    }
}, []) //eslint-disable-line
const handleAnimation = (mode) => {
    if (animation) {
        setAnimation(null)
    } else {
        setAnimation(mode)
    }
    // Jumpstart animation
    invalidate()
}

const handleGenerate = async () => {
    const prompt = promptRef.current.value
    if (prompt === "") {
        showToast("error", "Error", "Please enter a prompt", toastRef)
        return
    }
    if (isGenerating) {
        return
    }
    setIsGenerating(true)
    let api_key =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlODI3MTcsInVzZXI6IjE2NTQ3N2Y1MjgzNWExZjV  

mN2Y3YWVhZTcifQ.pA_jfnPaD_Rtw_p-Qqv05FwXJTLmfSrHD1DO7bQkzYw"
    let url = "https://api.wizmodel.com/sdapi/v1/txt2img"
    let payload = {
        "prompt": prompt,
    }
    let headers = {
        'Content-Type': 'application/json', 'Authorization': 'Bearer ' + api_key
    }
    let response = await fetch(url, {
        method: 'POST', headers: headers, body: JSON.stringify(payload)
    })
    let data = await response.json()
    setIsGenerating(false)
    setDialogVisible(false)
    try {
        setDecalPath("data:image/png;base64," + data.images[0])
    } catch (e) {
        showToast("error", "Error", "API KEY Expired", toastRef)
    }
    showToast("success", "Image Generation", "Image has been generated", toastRef)
}

const toastRef = useRef(null)
const promptRef = useRef(null)
const [dialogVisible, setDialogVisible] = useState(false)
const [isGenerating, setIsGenerating] = useState(false)
const footerContent = (<SavePopupFooter>
    <div className="btn-cancel">
        <Button label="Cancel" onClick={() => setDialogVisible(false)}/>
    </div>

```

```

<div className="btn-ok">
  <Button label="Generate" icon={isGenerating ? "pi pi-spin pi-spinner" : null} onClick={handleGenerate}/>
</div>
</SavePopupFooter>;

return (<div className={s.wrapper}>
  <Toast ref={toastRef}/>
  <Dialog visible={dialogVisible} modal header="Generate Image using AI" footer={footerContent}
    style={{width: '30rem', zIndex: 10000}}
    onHide={() => setDialogVisible(false)}>
    <SavePopup>
      <div>
        <label>
          Enter Prompt:
        </label>
        <input type="text" onKeyDown={() => setIsWriting(true)} onKeyUp={() => setIsWriting(false)}
          ref={promptRef}/>
      </div>
    </SavePopup>
  </Dialog>
  <div className={s.containerInner}>
    <Card
      id="color-container"
      title="Color"
      src="pi pi-palette"
      style={{width: "248px"}}
    >
      <Compact
        className={s.colorPicker}
        color={modelColor}
        onChange={(c) => {
          setModelColor(c.hex)
        }}
      />
    </Card>

    <Card
      id="background-container"
      title="Background"
      src="pi pi-slack"
      comp={bgData.map((bg) => {
        return (<Icon
          onClick={() => {
            setSet(bg.name)
          }}
          style={{width: '28px', height: '28px', backgroundImage: `url("${bg.thumb}")`} }
          match={set}
          id={bg.name}
          key={bg.name}
        />)
      })}
    >
    {set === "bg_color" && (<Compact
      className={s.colorPicker}
      color={backgroundColor}
      onChange={(c) => {
        setBackgroundColor(c.hex)
      }}
    )}
  </div>
</div>

```

```

    />})
    {(set === "bg_image" || set === "bg_transparent") && (<
      <div className={s.gridContainer}>
        <Icon
          imgSrc={"/thumbs/upload.svg"}
          key={"custom"}
          id={"custom"}
          match={backgroundImage.name}
          onClick={() => customBgRef.current.click()}
        />
        <input
          ref={customBgRef}
          style={{display: "none"}}
          type="file"
          accept="image/*"
          onChange={(data) => setBackgroundImage({
            data: data.target.files[0], name: "custom",
          })}
        />
        {bgSampleImages.map((img) => {
          return (<Icon
            imgSrc={img.path}
            key={img.name}
            id={img.name}
            match={backgroundImage.name}
            onClick={() => {
              setSet("bg_image")
              setBackgroundImage({
                path: img.path, name: img.name, author: img.author,
              })
            }}
          />
        )}
        <Icon
          imgSrc={"/thumbs/bg_transparent.svg"}
          onClick={() => {
            setSet('bg_transparent')
          }}
          match={set}
          id={'bg_transparent'}
        />
      </div>
    </>)}
  </Card>

  <Card
    id="misc-container"
    title="Misc"
    src="pi pi-prime"
  >
    <div className="flex justify-center items-center">
      <Icon
        title="Animation"
        onClick={() => handleAnimation("animation_360")}
        match={animation}
        id="animation_360"
        style={{margin: '10px', backgroundImage: 'url(/thumbs/rotate.svg)'}}
      />

```

```

<Icon
  title="Lights"
  onClick={() => setLights(!lights)}
  id="lights"
  imgSrc={lights ? '/thumbs/bulb_off.svg' : '/thumbs/bulb_on.svg'}
  style={{
    margin: '10px', backgroundColor: "white",
  }}
/>
<Icon
  title="Props"
  onClick={() => {
    props === "prop_shapes" ? setProps(null) : setProps("prop_shapes")
  }}
  match={props}
  id="prop_shapes"
  style={{ margin: '10px', backgroundImage: 'url(/thumbs/prop_shapes.png)' }}
/>
<Icon
  title="Image Gen AI"
  onClick={() => {
    setDialogVisible(true);
  }}
  match={props + "AI"}
  textContent={"AI"}
  style={{ margin: '10px' }}
/>
</div>
</Card>

<Card
  id="size-container"
  title="Size"
  src="pi pi-arrows-alt"
>
  <div className="flex justify-center items-center">
    <Icon
      title="Small"
      onClick={() => {
        setSizeType("size_S")
        setScale(0.95)
      }}
      match={sizeType}
      id="size_S"
      style={{ margin: '10px' }}
      textContent={"S"}
    />
    <Icon
      title="Medium"
      onClick={() => {
        setSizeType("size_M")
        setScale(1)
      }}
      match={sizeType}
      id="size_M"
      style={{ margin: '10px' }}
      textContent={"M"}
    />
  </div>

```

```

        <Icon
          title="Large"
          onClick={() => {
            setSizeType("size_L")
            setScale(1.05)
          }}
          match={sizeType}
          id="size_L"
          style={{ margin: '10px' }}
          textContent={"L"}
        />
        <Icon
          title="Extra Large"
          onClick={() => {
            setSizeType("size_XL")
            setScale(1.1)
          }}
          match={sizeType}
          id="size_XL"
          style={{ margin: '10px' }}
          textContent={"XL"}
        />
      </div>
    </Card>

    <Card
      id="decals-container"
      title="Decals"
      src="pi pi-th-large"
      style={{ gridColumn: 'span 2' }}
    >
      <DecalManager/>
    </Card>

    <Card
      id="text-container"
      title="Text"
      src="pi pi-pencil"
      style={{ gridColumn: 'span 2' }}
    >
      <TextEditor setDecalPath={setDecalPath}/>
    </Card>
  </div>
</div>
)
}

```

/components/hotkeys/Hotkey.js

```

import React from "react"
import IconWithText from "../iconWithText/IconWithText"
import useStore from "@states/modelState"
import s from "../hotkeys.module.css"
import styled from "styled-components";

const Hotkeys = () => {
  const { decalPath, animation } = useStore()
  return (

```

```

    <StyledKeysContainer className={s.wrapper} style={{ width: decalPath ? "200px" : "120px", display: (!decalPath &&
animation) ? "none" : null }}>
      {decalPath && (
        <
          <div className={s.inner}>
            <
              <IconWithText
                imgSrc={'/keys/keyUp.svg'}
                imgAlt="up"
                textContent="Scale up"
              />
              <IconWithText
                imgSrc={'/keys/keyDown.svg'}
                imgAlt="down"
                textContent="Scale down"
              />
              <IconWithText
                imgSrc={'/keys/keyEscRed.svg'}
                imgAlt="Esc"
                textContent="Cancel"
              />
            </>
          </div>
          { !animation && <hr style={{ height: '5px', color: 'white' }} /> }
        </>
      )}
    </StyledKeysContainer>
  )
}

```

export default Hotkeys

```

const StyledKeysContainer = styled.div`
  background-color: var(--primary-color-black);
`;

```

/components/card/Card.js

```

import s from "/card.module.css"
import { Button } from "primereact/button";

```

```

const Card = ({title, grid, children, src, style, comp, id}) => {
  return (
    <div className={s.container} style={style}
      id={id}>
      <div className={s.title}>
        <Button label={title} icon={src}/>
        {comp && <div className={s.comp}>{comp}</div>}
      </div>

```



```

        <div className={grid ? s.inner_grid : s.inner_block}>
          {children}
        </div>
      </div>
    )
  }

```

export default Card

/components/icon/Icon.js

import s from "../icon.module.css"

```

export default function Icon({
  imgSrc,
  imgAlt,
  textContent,
  onClick,
  id,
  match,
  inactive,
  ...rest
}) {
  const handleClick = () => onClick()

  return (
    <div
      onClick={inactive ? null : handleClick}
      className={inactive ? s.wrapper_inactive : s.wrapper}
      {...rest}
    >
      {imgSrc && <img className={s.icon} src={imgSrc} alt={imgAlt} />}
      {id === match && <div className={s.flag} />}
      {textContent && <p className={s.text}>{textContent}</p>}
    </div>
  )
}

```

/components/decalManager/DecalManager.js

```

import React, {useRef} from "react"
import {MdClose} from "react-icons/md"
import useStore from "../../states/modelState"
import s from "../decalManager.module.css"
import {FaPlus} from "react-icons/fa"
import {Ripple} from "primereact/ripple";
import useDecalStore from "@states/decalState";

```

```

const DecalManager = () => {
  const {decalImages, modelColor, setDecalImages, removeDecal} = useStore()
  const {decalsData, setDecalsData} = useDecalStore()

  const handleRemove = (event) => {
    const key = event.target.dataset.key
    let newArr = [...decalImages]
    let i = newArr.findIndex((el) => el.key === key) //eslint-disable-line
    newArr.splice(i, 1)
    setDecalImages(newArr)
    newArr = [...decalsData]
  }
}

```

```

    i = newArr.findIndex((el) => el.key === key) //eslint-disable-line
    newArr.splice(i, 1)
    removeDecal(key)
  }

  return (
    <div className={s.imagesWrapper}>
      <AddDecal/>
      {decalImages.map((decal, i) => {
        return (
          <div
            key={decal.key}
            className={s.imgContainerWrapper}
            style={{
              zIndex: decalImages.length - i,
            }}
          >
            <div className={s.imgContainer} title={decal.name}>
              <img
                className={s.img}
                src={decal.path}
                style={{ backgroundColor: modelColor, transition: "none" }}
                alt={"decal thumbnail"}
              />
            </div>
            <div
              className={s.closeBtn}
              onClick={handleRemove}
              data-key={decal.key}
            >
              <MdClose style={{ pointerEvents: "none" }}/>
            </div>
          </div>
        )
      })}
    </div>
  )
}

```

```
export default DecalManager
```

```

const AddDecal = () => {
  const inputRef = useRef()
  const {setDecalPath, setDecalName} = useStore()

  const loadDecal = () => {
    const name = inputRef.current.files[0].name
    const path = URL.createObjectURL(inputRef.current.files[0])
    const reader = new FileReader();

    reader.onload = function (event) {
      const base64String = event.target.result;
      setDecalPath(base64String)
      setDecalName(name)
    };
    reader.readAsDataURL(inputRef.current.files[0]);

    inputRef.current.value = ""
  }
}

```

```

    }

    const handleButtonClick = () => inputRef.current.click()
    return (
      <div className={s.addDecal} onClick={handleButtonClick}>
        <FaPlus/>
        <Ripple/>
      </div>
      <input
        style={{ display: "none" }}
        onInput={() => loadDecal()}
        ref={inputRef}
        type="file"
        accept="image/*"
      />
    </>
  )
}

```

/components/TextEditor.js

```

export default function TextEditor({setDecalPath}) {
  const { incrementDecalSize, setIsWriting } = useStore()
  const editorRef = React.useRef(null);
  const [content, setContent] = useState("");

  const formats = [
    'header', 'bold', 'italic', 'underline', 'strike',
    'color', 'background', 'font', 'align', 'clean'
  ];

  const modules = {
    toolbar: [
      [{ 'header': [1, 2, 3, false]}, { 'font': ["sans-serif", "serif", "monospace"]}],
      [{ 'color': []}, { 'background': []}, { 'align': []}],
      ['bold', 'italic', 'underline', 'strike', 'clean']
    ]
  };

  const handleEditorChange = (newContent) => {
    setContent(newContent);
  };

  const convertToImage = () => {
    const editor = editorRef.current;
    const quill = editor.getEditor();
    quill.root.style.width = 'fit-content';
    const contentWidth = quill.root.scrollWidth;
    const newWidth = contentWidth + 10; // Adjust the value as needed
    quill.root.style.width = newWidth + 'px';
    htmlToImage.toPng(quill.root)
      .then(function (dataUrl) {
        incrementDecalSize(0.2);
        setDecalPath(dataUrl);
        quill.root.style.width = 'auto';
      })
      .catch(function (error) {
        console.error("Conversion failed:", error);
      });
  };
}

```

```

    });
};

return (
  <>
    <ReactQuill
      id="text-format-container"
      theme="snow"
      value={content}
      onKeyDown={() => setIsWriting(true)}
      onKeyUp={() => setIsWriting(false)}
      onChange={handleEditorChange}
      modules={modules}
      formats={formats}
      forwardedRef={editorRef}
    />
    <button className={"add-text-btn"} onClick={convertToImage}>
      <MdAddBox />
    </button>
  </>
);
}

```

/create/recent/shirt/[id]/page.jsx

```

"use client"
import React, { useEffect, useState } from 'react'
import useStore from "@/states/modelState";
import useAppStore from "@/states/appState";
import useDecalStore from "@/states/decalState";
import { LoadingScreen } from "@/styles/styledCreate";
import Studio from "@/components/studio/Studio";

const Page = ({ params }) => {
  const {
    setModelColor,
    setBackgroundColor,
    setScale,
    setSizeType,
  } = useStore()
  const { setDecalsData } = useDecalStore()
  const { saved, setSaved } = useAppStore()

  const [loading, setLoading] = useState(true)

  useEffect(() => {
    fetch('/api/get/model?id=' + params.id)
      .then(res => res.json())
      .then(bigData => {
        const { _doc: data } = bigData
        setModelColor(data.modelColor)
        setBackgroundColor(data.backgroundColor)
        setScale(data.scale)
        setSizeType(data.sizeType)
        setDecalsData(data.decalsData)
        setLoading(false)
        setSaved({ ...saved, name: data.name, desc: data.desc, isUnSaved: false, state: true })
      })
  })
}

```

```

    })
  }, []
)

return (
  <
    {
      loading ? (
        <LoadingScreen>
        <div>
          
        </div>
        </LoadingScreen>
      ) : <Studio mode={ "recent" } modelUrl={'/models/shirt/tshirt.glb'}/>
    }
  </>
)
}
export default Page

```

/api/get/model/route.js

```

import { NextResponse } from "next/server";
import connect from "@/lib/db";
import Model from "@/models/Model";

export const dynamic = 'force-dynamic'

export const GET = async (req) => {
  try {
    const { searchParams } = req.nextUrl
    const id = searchParams.get('id');
    await connect()
    const model = await Model.findOne({ _id: id })
    return NextResponse.json({ ...model });
  } catch (error) {
    console.log(error);
    return NextResponse.json({ message: 'Error Saving Model: ' + error });
  }
}

```

/api/get/recent-models/route.js

```

import { NextResponse } from "next/server";
import connect from "@/lib/db";
import Model from "@/models/Model";

export const dynamic = 'force-dynamic'

export const GET = async (req) => {
  try {
    const { searchParams } = req.nextUrl
    const email = searchParams.get('user');
    await connect()
    const models = await Model.find({ email }).sort({ createdAt: -1 }).limit(4);
    if (typeof models === 'object')
      return NextResponse.json({ models: models });
    return NextResponse.json({ models: models });
  } catch (error) {
    console.log(error);
  }
}

```

```

    return NextResponse.json({message: 'Error Saving Model: ' + error});
  }
}

```

/api/get/user/profile/route.js

```

import {NextResponse} from "next/server";
import connect from "@/lib/db";
import User from "@/models/User";

export const POST = async (req) => {
  const {email} = await req.json();

  try {
    await connect()
    const currentUser = await User.findOne().where({email: email});
    return NextResponse.json({currentUser: currentUser});
  } catch (error) {
    console.log(error);
    return NextResponse.json({message: 'Error Creating Account: ' + error});
  }
}

```

/api/post/save-model/verify/route.js

```

import {NextResponse} from "next/server";
import connect from "@/lib/db";
import Model from "@/models/Model";

export const POST = async (req) => {
  const {name, email} = await req.json();
  try {
    await connect()
    const model = await Model.find({name, email})
    if (model.length > 0) {
      return NextResponse.json({exists: true});
    }
    return NextResponse.json({exists: false});
  } catch (error) {
    console.log(error);
    return NextResponse.json({message: 'Error Saving Model: ' + error});
  }
}

```

/api/post/save-model/route.js

```

import {NextResponse} from "next/server";
import connect from "@/lib/db";
import Model from "@/models/Model";
export const POST = async (req) => {
  const {savedObj} = await req.json();
  const {email,backgroundColor, modelType, decalsData, modelColor, name, desc, sizeType, scale, snapshot} = savedObj
  try {
    await connect()
    const data = await Model.find().where({name: name, email: email})

    if (data.length > 0) {
      await Model.updateOne({
        name: name,
        email: email,

```

```

    }, {
      scale: scale, snapshot: snapshot, sizeType: sizeType, modelType: modelType, decalsData: decalsData, modelColor:
modelColor, backgroundColor: backgroundColor
    })
  } else {
    await Model.create({
      email: email, name: name, desc: desc, scale: scale, snapshot: snapshot, sizeType: sizeType, modelType: modelType,
decalsData: decalsData, modelColor: modelColor, backgroundColor: backgroundColor
    })
  }

  return NextResponse.json({message: 'Model was Saved'});
} catch (error) {
  console.log(error);
  return NextResponse.json({message: 'Error Saving Model: ' + error});
}
}

```

/api/post/send-feedback/route.js

```

import {NextResponse} from "next/server";
import connect from "@/lib/db";
import Review from "@/models/Review";

export const POST = async (req) => {
  const {savedFeedback} = await req.json();

  const {firstName, lastName, email, feedback} = savedFeedback;
  try {
    await connect()

    await Review.create({
      firstName: firstName,
      lastName: lastName,
      email: email,
      feedback: feedback,
    })

    return NextResponse.json({message: 'Review sent'});
  } catch (error) {
    console.log(error);
    return NextResponse.json({message: 'Error Sending Review: ' + error});
  }
}

```

/api/post/user/login/route.js

```

import {NextResponse} from "next/server";
import connect from "@/lib/db";
import User from "@/models/User";

export const POST = async (req) => {
  const {user_email, user_password} = await req.json();
  const thirtyDaysInSeconds = 30 * 24 * 60 * 60; // 30 days in seconds

  const cookieOptions = {
    'Max-Age': thirtyDaysInSeconds,
    path: '/',
  };
}

```

```

try {
  await connect();
  const user = await User.findOne({
    email: user_email,
  });
  if (user) {
    return new Response(JSON.stringify({...user, found: true}), {
      headers: {
        "Content-Type": "application/json",
        "Set-Cookie": `session=true; Max-Age=${cookieOptions['Max-Age']}; path=${cookieOptions.path}`,
      },
      status: 200
    });
  } else {
    return new Response(JSON.stringify({...user, found: false}), {
      headers: {
        "Content-Type": "application/json",
        "Set-Cookie": `session=false; Max-Age=${cookieOptions['Max-Age']}; path=${cookieOptions.path}`,
      },
      status: 200
    });
  }
} catch (error) {
  console.log(error);
  return NextResponse.json({message: 'Error Logging In: ' + error});
}
}

```

/api/post/user/signup/route.js

```

export const POST = async (req) => {
  const {signupUserDetails, found} = await req.json();
  const {signupEmail, signupPassword, signupUsername, signupPhone} = signupUserDetails;
  const thirtyDaysInSeconds = 30 * 24 * 60 * 60; // 30 days in seconds
  const cookieOptions = {
    'Max-Age': thirtyDaysInSeconds,
    path: '/',
  };
  try {
    await connect()
    const user = await User.create({
      email: signupEmail,
      password: signupPassword,
      username: signupUsername,
      phone: signupPhone,
    })
    return new Response(JSON.stringify({user, message: "Account Created"}), {
      headers: {
        "Content-Type": "application/json",
        "Set-Cookie": `session=true; Max-Age=${cookieOptions['Max-Age']}; path=${cookieOptions.path}`,
      },
      status: 201
    });
  } catch (error) {
    console.log(error);
    return NextResponse.json({message: 'Error Creating Account: ' + error});
  }
}

```



```
}  
}
```

/lib/db.js

```
import mongoose from 'mongoose'  
  
export default async function connect() {  
  if (mongoose.connections[0].readyState) {  
    return;  
  }  
  try {  
    await mongoose.connect(process.env.MONGO_URI);  
    console.log("Connected to MongoDB");  
  }  
  catch (error) {  
    console.log(error);  
    console.log("Failed to connect to MongoDB");  
    throw error  
  }  
}
```

/lib/driver.js

```
import {driver} from "driver.js";  
import "../styles/driver.css";  
  
export const navDriver = () => {  
  const driverObj = driver({  
    showProgress: true, steps: [{  
      element: '#project-link', popover: {  
        title: 'SnX',  
        description: 'Explore SnX and customize your outfit. Dive into a world of personalized fashion where you can tailor  
your look to match your unique style and preferences.'  
      }  
    }, {  
      element: '#home-link', popover: {  
        title: 'Home',  
        description: 'Return to the Home page. Navigate back to the main landing page to explore new features, updates, and  
announcements related to SnX.'  
      }  
    }, {  
      element: '#create-link', popover: {  
        title: 'Create',  
        description: 'Initiate the creation of new outfits. Get started on designing and crafting your latest ensemble with our  
intuitive outfit customization tools.'  
      }  
    }, {  
      element: '#contact-link', popover: {  
        title: 'Contact',  
        description: 'Connect with customer support. Reach out to our dedicated customer service team for assistance, inquiries,  
or feedback regarding your SnX experience.'  
      }  
    }, {  
      element: '#help-button', popover: {  
        title: 'Guide',  
        description: 'Access the user guide. Explore our comprehensive guide designed to help you navigate through SnX\'s  
features and functionalities seamlessly.'  
      }  
    }  
  ]  
});
```

```

    }, {
      element: '#account-link', popover: {
        title: 'Profile',
        description: 'View and manage your profile. Access your personalized profile dashboard to track your activity, manage preferences, and update account details effortlessly.'
      }
    }]
  });

  driverObj.drive();
}

export const createDriver = () => {
  const driverObj = driver({
    showProgress: true,
    steps: [{
      element: '#shirt-button',
      popover: {title: 'Shirt', description: 'Customize a shirt'}
    }, {
      element: '#shoes-button',
      popover: {title: 'Shoes', description: 'Design a shoes now'}
    }, {
      element: '#pant-button',
      popover: {title: 'Pant', description: 'Create a new Pant'}
    }, {
      element: '#cap-button',
      popover: {title: 'Cap', description: 'Start designing a cap'}
    }, {element: '#recents', popover: {title: 'Recents', description: 'Check out your recent creations'}},]
  });

  driverObj.drive();
}

export const createDesignerDriver = () => {
  const driverObj = driver({
    showProgress: true, steps: [{
      element: '#studio-viewer', popover: {
        title: 'Model',
        description: 'Design your model as you wish. Use various tools and options to create a unique and personalized design.'
      }
    }, {
      element: '#design-options', popover: {
        title: 'Options',
        description: 'Choose from a variety of design options. Explore different features and functionalities to enhance your model.'
      }
    }, {
      element: '#color-container', popover: {
        title: 'Colors',
        description: 'Select colors for your model. Choose from a wide range of color options to customize your design.'
      }
    }, {
      element: '#background-container', popover: {
        title: 'Set Background',
        description: 'Set the background color or image for your model. Personalize the backdrop to complement your design.'
      }
    }, {
      element: '#misc-container', popover: {

```

```

        title: 'Miscellaneous',
        description: 'Access miscellaneous settings for your model. Fine-tune various parameters to achieve your desired look.'
    }, {
        element: '#size-container', popover: {
            title: 'Sizes',
            description: 'Select the size of your model. Choose the dimensions that best fit your requirements and preferences.'
        }
    }, {
        element: '#decals-container', popover: {
            title: 'Decals',
            description: 'Add decals and images to your model. Enhance your design with custom decals and graphics.'
        }
    }, {
        element: '#text-container', popover: {
            title: 'Text',
            description: 'Add custom text to your model. Personalize your design with unique text elements and messages.'
        }
    }, {
        element: '#text-format-container', popover: {
            title: 'Text Format',
            description: 'Type your text and apply styling. Customize the appearance and formatting of your text elements.'
        }
    }, {
        element: '#reset-button', popover: {
            title: 'Reset',
            description: 'Reset your model to its default state. Undo any changes and start fresh with your design.'
        }
    }, {
        element: '#export-container', popover: {
            title: 'Export',
            description: 'Export your model as an image or 3D model file. Save and share your designs with others.'
        }
    }, {
        element: '#save-button', popover: {
            title: 'Save',
            description: 'Save your model. Preserve your design for future reference or sharing with others.'
        }
    }
  ]
});

driverObj.drive();
}

export const contactDriver = () => {
  const driverObj = driver({
    showProgress: true, steps: [{
      element: '#about-us', popover: {
        title: 'About Us',
        description: 'Learn more about our company and what we offer. Customize a shirt to your liking.'
      },
    }, {
      element: '#faq', popover: {
        title: 'FAQ',
        description: 'Find answers to frequently asked questions. Design shoes now and explore our FAQ section.'
      },
    }, {
      element: '#contact-links', popover: {

```

```

        title: 'Contact Links',
        description: 'Connect with us through various contact channels. Create a new pant and get in touch.'
      },
    ], {
      element: '#recents', popover: {
        title: 'Recents',
        description: 'View your recent creations and explore your design history. Check out your recent designs.'
      },
    }
  ]
});

driverObj.drive();
}

export const accountDriver = () => {
  const driverObj = driver({
    showProgress: true, steps: [{
      element: '#delete-account',
      popover: {title: 'Delete Account', description: 'Click here to delete your account.'}
    }, {
      element: '#username_text',
      popover: {title: 'Username', description: 'Your username is displayed here.'}
    }, {
      element: '#email_text',
      popover: {title: 'Email', description: 'Your email address is shown here.'}
    }, {element: '#password_text', popover: {title: 'Password', description: 'Your password is displayed here.'}}, {
      element: '#phone_text',
      popover: {title: 'Phone Number', description: 'Your phone number is displayed here.'}
    },
  ],
});

  driverObj.drive();
}

```

/globals.css

```

@tailwind base;
@tailwind components;
@tailwind utilities;

:root {
  --primary-theme-color: #008080;
  --primary-toast-color: #008080AA;
  --primary-comp-bg: #2c2c2c;
  --primary-bg: rgba(20, 20, 20, 0.75);
  --primary-text-color: #ccc;
}

body {
  margin: 0;
  padding: 0;
  width: 100vw;
  height: 100svh;
  overflow-x: hidden;
  font-family: sans-serif;
  background-color: #1d1d1d;
}

```

```

.container {
  display: flex;
  height: 100%;
}

.showAuth {
  opacity: 1 !important;
}

.hideAuth {
  opacity: 0.05 !important;
}

.shadow-md {
  box-shadow: none !important;
}

.ql-editor {
  background: transparent;
  height: 70px !important;
  max-height: 100px;
  overflow: scroll;
}

.ql-container {
  border: none !important;
}

.add-text-btn {
  position: absolute;
  top: 1px;
  right: 2px;
  color: var(--primary-theme-color);
  font-size: 40px;
}

.ql-picker.ql-font .ql-picker-item {
  font-size: 0;
}

.ql-picker.ql-font .ql-picker-item:before {
  content: attr(data-value) !important;
  font-size: 14px;
}

.ql-picker.ql-font .ql-active:before {
  content: attr(data-value) !important;
  font-size: 14px;
}

.export-glb-div {
  display: flex;
  justify-content: space-between;
  align-items: center;
  position: absolute;
  bottom: 20px;
}

```

```

    right: 30px;
    gap: 20px;
    flex-direction: row-reverse;
    z-index: 10;
  }

.export-glb-div > button,
.export-glb-div > div > button {
  padding: 10px;
  background: var(--primary-comp-bg);
}

.export-glb-div > button:hover,
.export-glb-div > div:hover {
  background: var(--primary-theme-color);
}

.export-glb-div > div > button {
  box-shadow: none !important;
}

.export-glb-div > div > button:hover {
  background: var(--primary-theme-color);
}

.export-glb-div > div > button.p-splitbutton-menubutton {
  width: 2rem !important;
}

.p-toast-message-icon {
  display: none;
}

.p-toast-message-content {
  padding: 10px;
  align-items: center;
}

.p-toast-message {
  border-style: solid;
  border-width: 0 0 0 6px;
  border-radius: 0;
  box-shadow: 0 0 1rem rgba(0, 0, 0, 0.25);
  backdrop-filter: blur(16px) saturate(180%);
  -webkit-backdrop-filter: blur(16px) saturate(180%);
}

```

/styles/styledCreate.js

```

import { styled } from "styled-components";
import Link from "next/link";
import { Canvas } from "@react-three/fiber";
import { motion } from "framer-motion";

export const CardGrid = styled.div`
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 1rem;

```

```

position: relative;

> .recent-loader {
  width: 100%;
  height: 100%;
  grid-column: span 2;
  grid-row: span 2;
  background: crimson;
}
`;

export const StyledBlock = styled.div`
  border-radius: 0;
  background: var(--primary-comp-bg);
  //padding: 1rem;
  display: flex;
  justify-content: center;
  align-items: center;
  width: 200px;
  height: 200px;
  transition: all 0.2s ease;
  cursor: pointer;

  &:active {
    transform: scale(0.95);
  }

  &:hover {
    box-shadow: var(--primary-theme-color) 0 0 5px 3px;
  }

  > img {
    width: 100%;
    height: 100%;
    object-fit: cover;
  }

  > .inactive {
    filter: invert(7%) sepia(17%) saturate(2%) hue-rotate(314deg) brightness(98%) contrast(89%);
    width: 55%;
    height: 55%;
  }
`;

export const BlockIcon = styled(Link)`
  color: var(--primary-theme-color);
  width: 100%;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  transition: 0.25s all linear;

  > svg {
    width: 65%;
    height: 65%;
  }
`;

```

```

export const CardGridHeader = styled.div`
  position: absolute;
  top: 0;
  translate: 0 -100%;
  font-size: 2.5rem;
  color: var(--primary-theme-color);
`;

export const CreateCanvasWrapper = styled(Canvas)`
  > div > canvas {
    height: 100vh !important;
    width: calc(100%);
  }
`;

export const SavingScreen = styled.div`
  position: absolute;
  inset: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  mix-blend-mode: color-burn;
  z-index: 1000;

  > div.save-loader {
    background: white;

    > img {
      width: 100%;
    }
  }
`;

export const LoadingScreen = styled.div`
  position: absolute;
  inset: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  mix-blend-mode: color-burn;
  z-index: 1000;

  > div.save-loader {
    background: white;

    > img {
      width: 100%;
    }
  }
`;

export const LoadingOverlay = styled.div`
  position: absolute;
  inset: 0;
  background: rgba(0, 128, 128, 0.4);
  z-index: 1001;

```



```

`
export const SavePopup = styled.div`
  margin-left: -1.25rem;
  padding-top: 20px;
  padding-inline: 10px;

  > p {
    font-family: "Bahnschrift", sans-serif;
    font-size: 1.25rem;
  }

  > div {
    display: grid;
    grid-template-columns: 35% 65%;
    gap: 5px;
    font-size: 1rem;
  }
`;

export const SavePopupFooter = styled.div`
  display: flex;
  width: 100%;
  gap: 10px;
  flex-direction: row-reverse;

  > .btn-cancel {
    border: 1px solid var(--primary-theme-color);
  }

  > .btn-ok {
    background: var(--primary-theme-color);
  }

  > div {
    padding: 5px;
    border-radius: 5px;
  }

  > div > button {
    outline: none !important;
  }
`;

```

5.5 Conclusion

This chapter consists of algorithm flowchart & coding was shown in this chapter. By seeing this chapter our project is easily understandable. As technology continues to advance, the project represents an innovative approach to addressing the complexities of the modern job market, ultimately contributing to individual success and societal progress.

Chapter 6: Testing and Debugging

6.1. Introduction

6.2. Testing Approach

6.3. Test Plan

6.4. Debugging Approach

6.5. Test cases

6.6. Conclusion

6.1. Introduction

The testing and debugging approach for the project is crucial for ensuring the system's reliability and accuracy. This approach encompasses systematic processes for identifying and rectifying software defects and inconsistencies that could affect performance. The introduction outlines key objectives, including verifying AI algorithms, validating user inputs and outputs, and maintaining data integrity, preparing and executing test cases. It also defines the scope of testing activities through test plan, clarifies roles and responsibilities, and addresses risk management considerations. By providing a clear framework for testing and debugging, the project aims to ensure the quality and effectiveness of the system throughout its development lifecycle.

6.2. Testing Approach

A test approach is a project's testing implementation strategy that defines the strategy which needs to be implemented and executed, to carry out a particular task. The test approach defines the testing methodology, tools, techniques, and strategies that will be used to test the software application. The testing approach also includes the roles and responsibilities of the testing team, the test environment, and the test data. The methods implemented in testing are discussed in brief. The reliability of the system cannot be ensured if the appropriate methods are not chosen to perform the test.

Unit Testing:

Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements. Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.

- Reduces Defects in the Newly developed features or reduces bugs when changing the
- existing functionality.
- Reduces Cost of Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well

Functional Testing:

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on simulation of actual system usage but does not develop any system structure assumptions. It is basically defined as a type of testing which verifies that each function of the software application works in conformance with the requirement and specification. Each functionality of the software

application is tested by providing appropriate test input, expecting the output and comparing the actual output with the expected output.

1. Determine which functionality of the product needs to be tested. This can vary from testing main functions, messages, error conditions and/or product usability.
2. Create input data for functionalities to be tested according to specified requirements.
3. Determine acceptable output parameters according to specified requirements.
4. Execute test cases.
5. Compare actual output from the test with the predetermined output values. This reveals if the system is working as expected.

Functionality testing plays a crucial role in software development and is an essential part of the software testing process. It helps to identify and fix defects and ensures that the software meets the needs of its intended users.

6.3. Test Plan

A test plan is a document that outlines the testing approach, scope, objectives, and deliverables of a software testing project. It is a comprehensive document that details the testing strategy, including the resources, timelines, and responsibilities for executing the tests. The plan typically contains a detailed understanding of the eventual workflow. A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from test engineers. A test plan is a crucial document that outlines the testing strategy and approach for a software testing project. It helps to ensure that the testing effort is well-organized, efficient, and effective, and that the software meets the requirements and quality standards. There are three major elements that should be described in the test plan: test coverage, test methods, and test responsibilities:

Test coverage - Test coverage in the test plan states what requirements will be verified during what stages of the product life. Test coverage is derived from design specifications and other requirements, such as safety standards or regulatory codes, where each requirement or specification of the design ideally will have one or more corresponding means of verification.

Test method - Test methods in the test plan state how test coverage will be implemented. Test methods may be determined by standards, regulatory agencies, or contractual agreement, or may have to be created new. Test methods also specify test equipment to be used in the performance of the tests and establish pass/fail criteria. Test methods used to verify hardware design requirements can range from very simple steps, such as visual inspection, to elaborate test procedures that are documented separately. **Test responsibilities** - Test responsibilities include what organizations will perform the test methods and at each stage of the product life. This allows test organizations to plan, acquire or develop test equipment and other resources necessary to implement the test

methods for which they are responsible. Test responsibilities also include what data will be collected and how that data will be stored and reported.

6.4. Debugging Approach

Debugging occurs as a consequence of successful testing. Debugging is an action that results in the removal of the error. It should be an orderly process. The external manifestation of the error and the internal cause of the error may have no obvious relationship to one another. Debugging will always have one of two outcomes:

1. The cause will be found and corrected.
2. The cause will not be found. In the other case, the person performing debugging may suspect a cause, design one or more test cases to help validate that suspicion and work toward error correction in an iterative action.

1. Brute Force Method

This is the foremost common technique of debugging however is that the least economical method. during this approach, the program is loaded with print statements to print the intermediate values with the hope that a number of the written values can facilitate to spot the statement in error. This approach becomes a lot of systematic with the utilization of a symbolic program (also known as a source code debugger), as a result of values of various variables will be simply checked and breakpoints and watch-points can be easily set to check the values of variables effortlessly.

2. Backtracking

This is additionally a reasonably common approach. during this approach, starting from the statement at which an error symptom has been discovered, the source code is derived backward till the error is discovered. sadly, because the variety of supply lines to be derived back will increase, the quantity of potential backward methods will increase and should become unimaginably large so limiting the utilization of this approach.

3. Cause Elimination Method

In this approach, a listing of causes that may presumably have contributed to the error symptom is developed and tests are conducted to eliminate every error. A connected technique of identification of the error from the error symptom is that the package fault tree analysis.

4. Program Slicing

This technique is analogous to backtracking. Here the search house is reduced by process slices. A slice of a program for a specific variable at a particular statement is that the set of supply lines preceding this statement which will influence the worth of that variable.

6.5. Test Cases

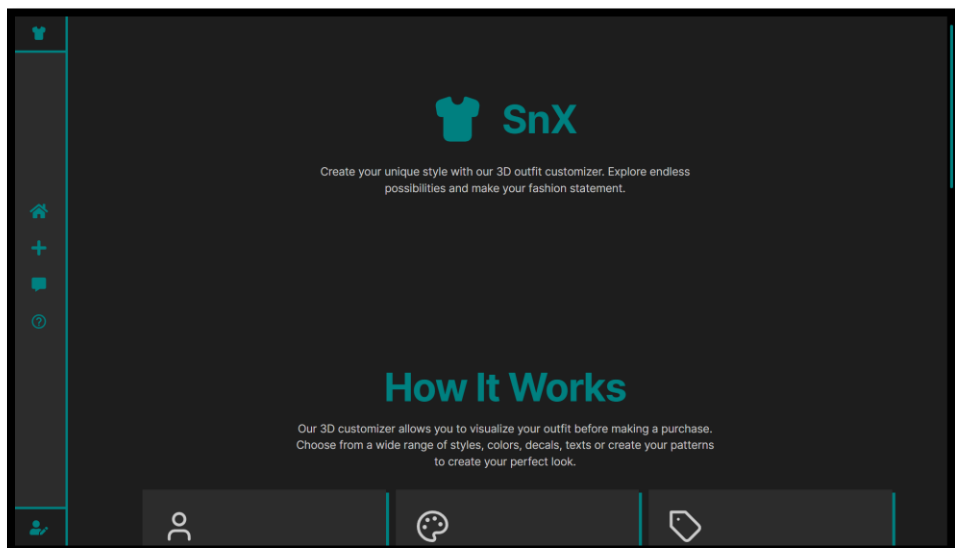
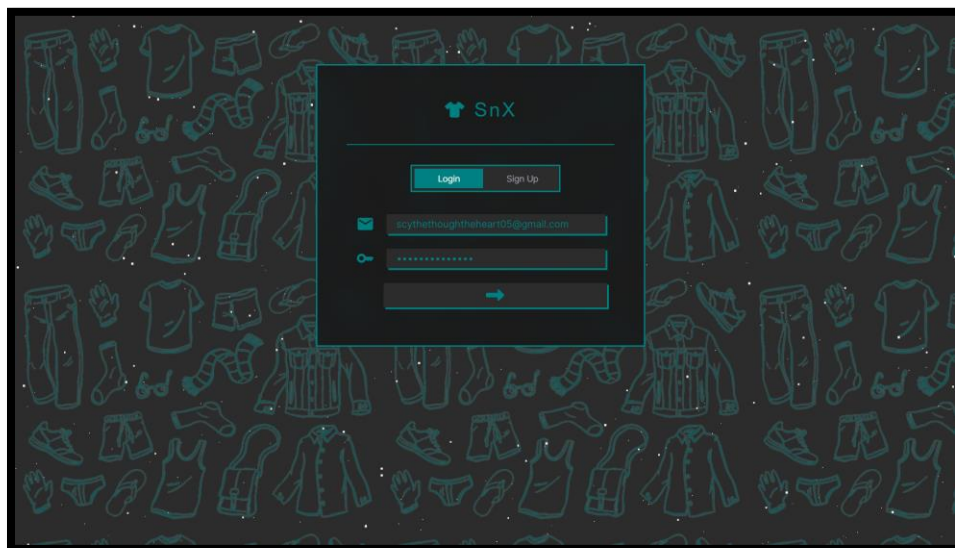
TEST CASES							
TC ID	Test Case Description	Precondition	Steps	I/P Data	Expected Result	Actual Result	Status
T1	Verify login using valid credentials	Login form should be active	1. Enter Credentials 2. Click Login	Valid Credentials	User should be logged in	User is logged in	Pass
T2	Verify login using invalid credentials	Login form should be active	1. Enter Credentials 2. Click Login	Invalid Credentials	User shouldn't be logged in	User isn't logged in	Pass
T3	Verify whether decals can be added from file dialog	Create Page should be active	1. Click on Add Decals 2. Choose from the file Dialog	Images	Image should be accepted	Image is accepted	Pass
T4	Verify whether text can be added to models	Create Page should be active	1. Type text in Quill 2. Click on Add Text button	Text	Decal Helper with the typed text should be available	Decal Helper with the typed text is available	Pass
T5	Verify whether AI can generate decals	Create Page should be active	1. Type prompt in AI Prompt Box 2. Click on Generate button	Text	Decal Helper with the generated image should be available	Decal Helper with the generated image is available	Pass
T6	Verify whether the reset button works as expected	Create Page should be active	1. Click on reset button	N/A	All the controls should reset to default	All the controls are reset to default	Pass
T7	Verify whether the save button works as expected	Create Page should be active	1. Click on save button	N/A	Dialog box should open and model should be saved	Dialog box opened and model is saved	Pass
T8	Verify whether the export function works as expected	Create Page should be active	1. Click on save button	N/A	Canvas should be exported as png / glb	Canvas is exported as png / glb	Pass
T9	Verify whether recently saved models can be opened	Recent Model Page should be active	1. Go to create page 2. Click on recently saved model	N/A	Model should be opened with the saved formatting	Model is opened with the saved formatting	Pass
T10	Verify session management	User should be logged in	1. Close the app 2. Open it again	N/A	If user is logged it, home page should be displayed	If user is logged it, home page is displayed	Pass

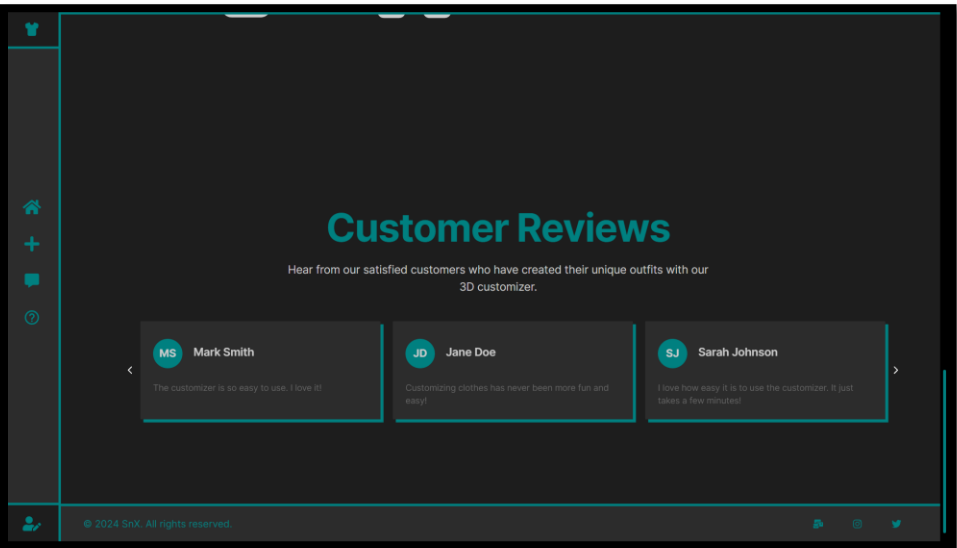
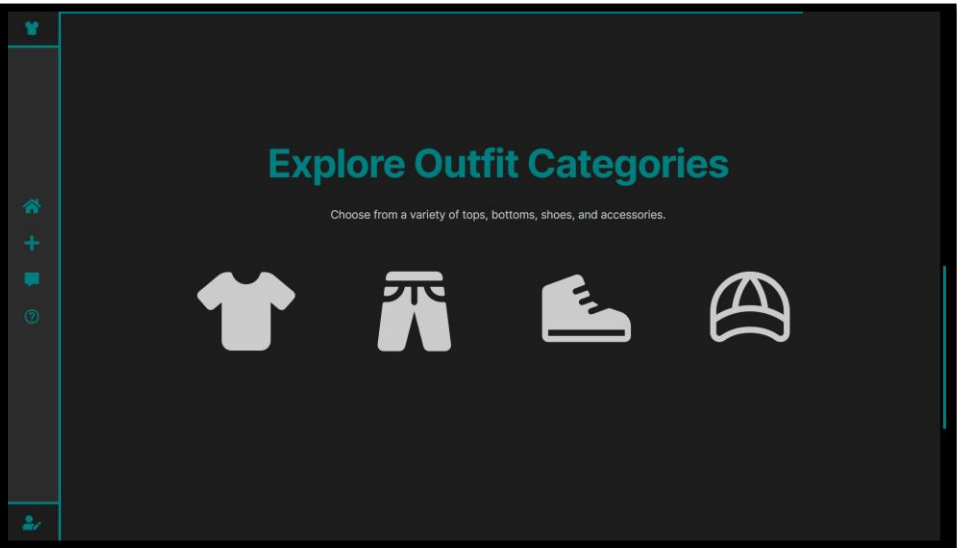
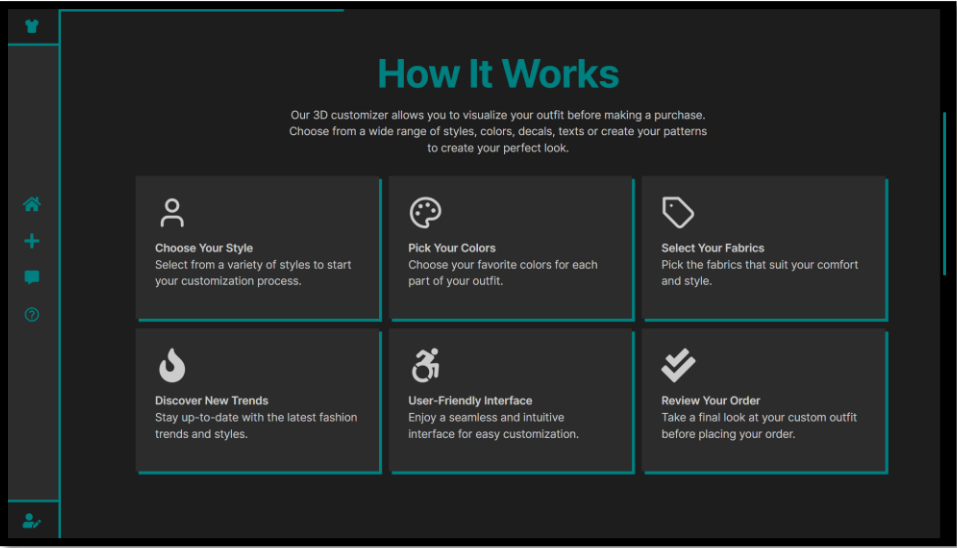
6.6. Conclusion

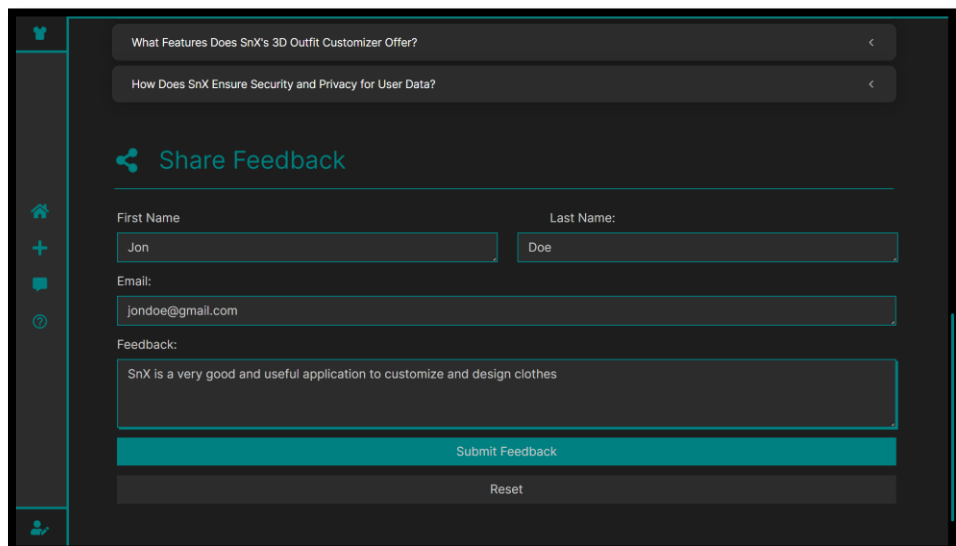
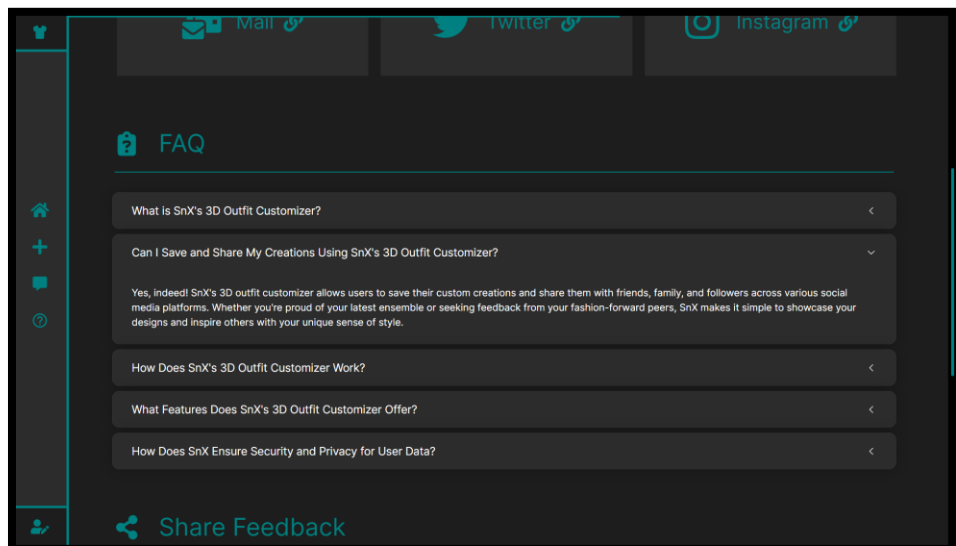
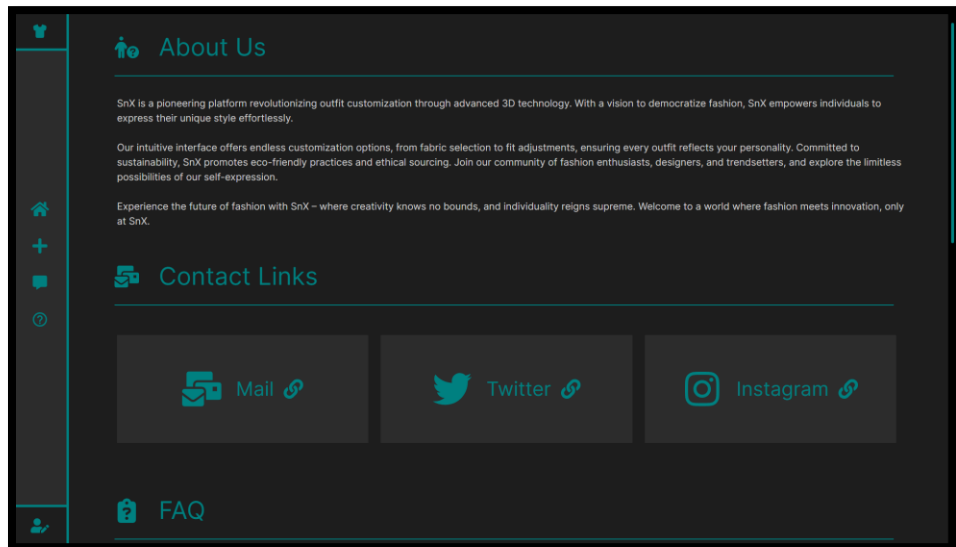
In conclusion, the testing and debugging approach for the project plays a critical role in guaranteeing the reliability, accuracy, and functionality of the system. Through systematic testing processes and diligent debugging efforts, potential software defects and inconsistencies are identified, isolated, and resolved to maintain the system's performance standards. The approach emphasizes the importance of decal algorithms, validating user inputs and outputs, and ensuring data integrity. Additionally, clear roles and responsibilities, along with proactive risk management strategies, contribute to effective testing execution and oversight. By prioritizing thorough testing and debugging practices, the project aims to deliver a robust and dependable solution that meets the needs of its users and stakeholders.

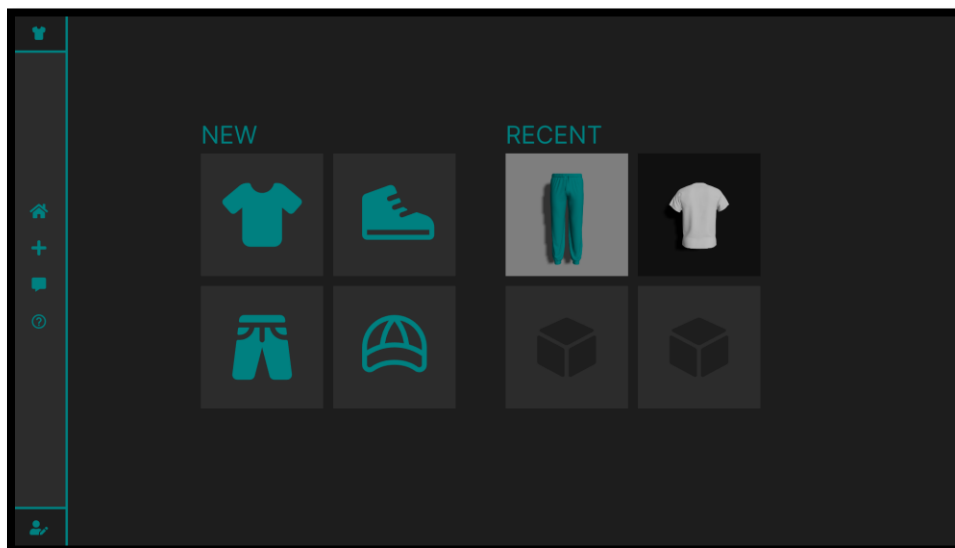
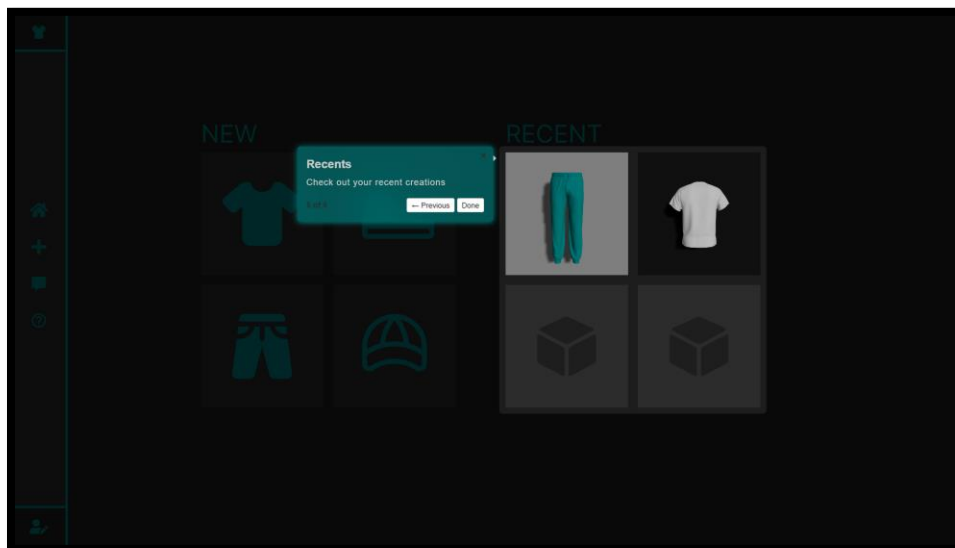
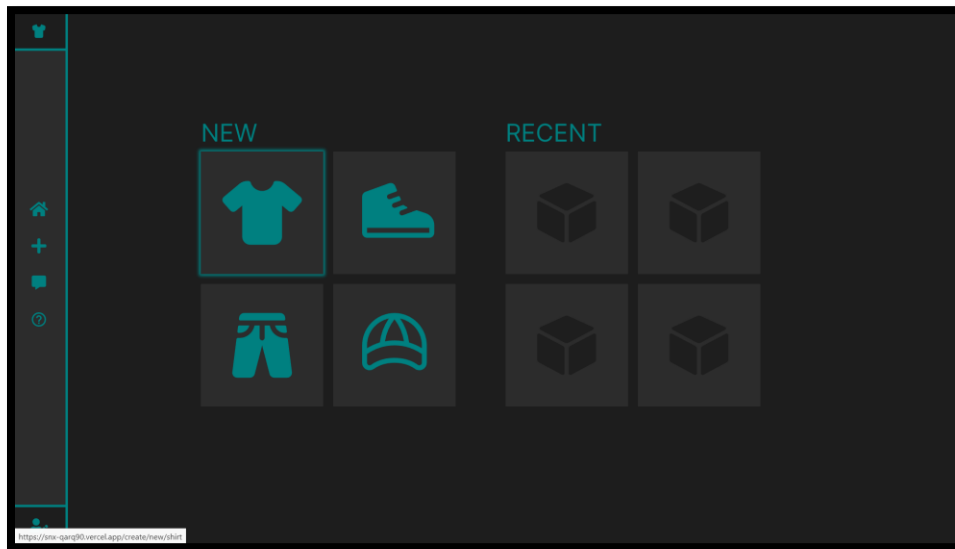
Chapter 7: Snapshots

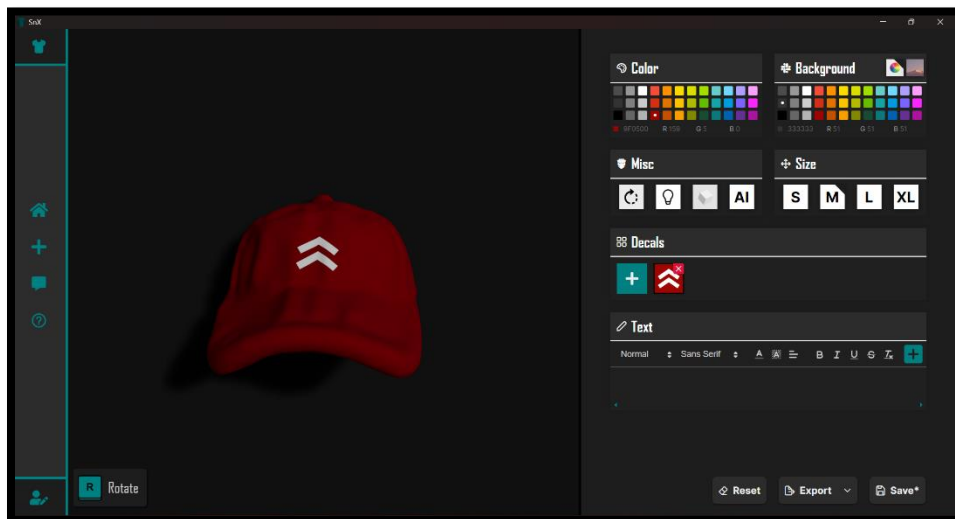
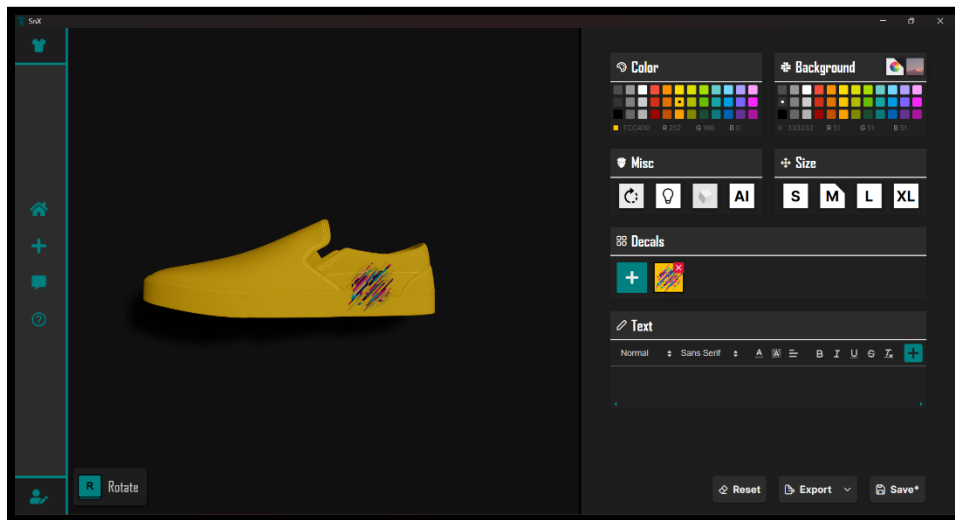
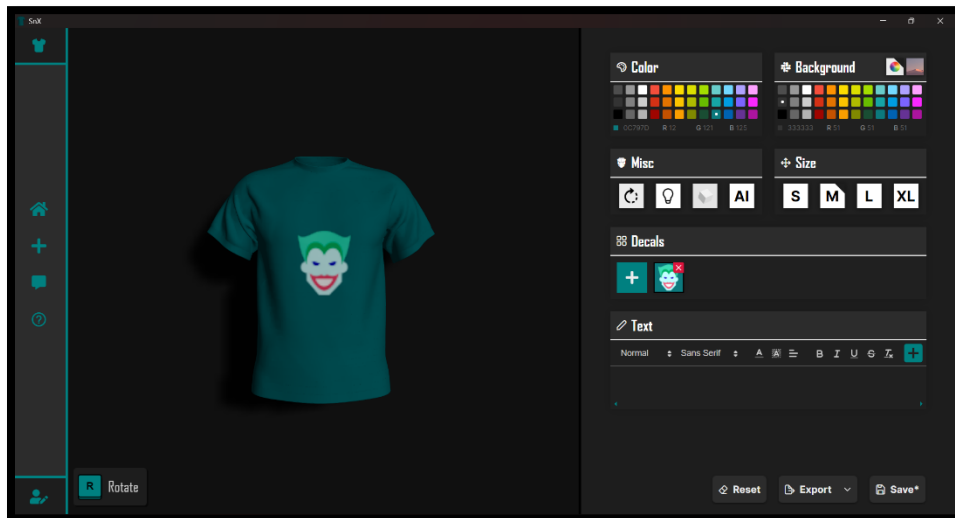
7.1 Snapshots

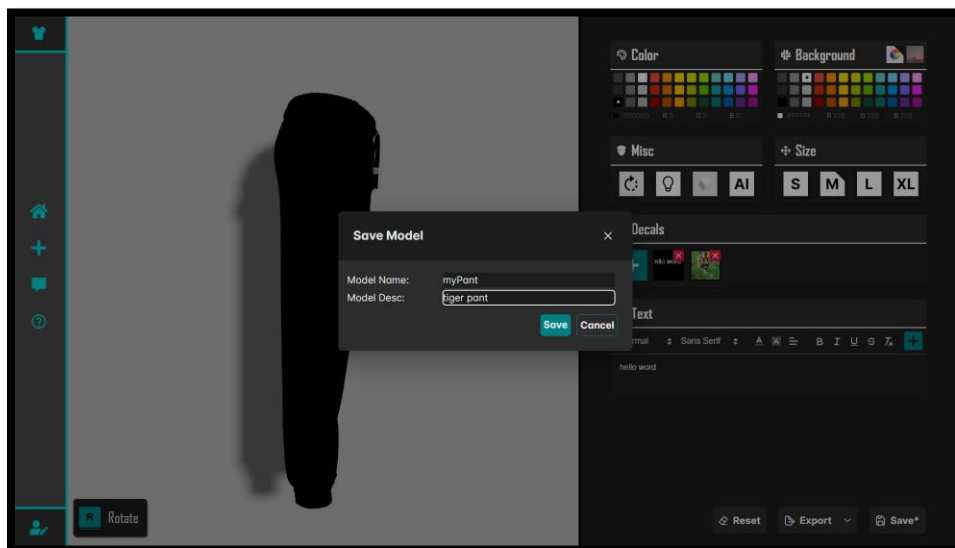
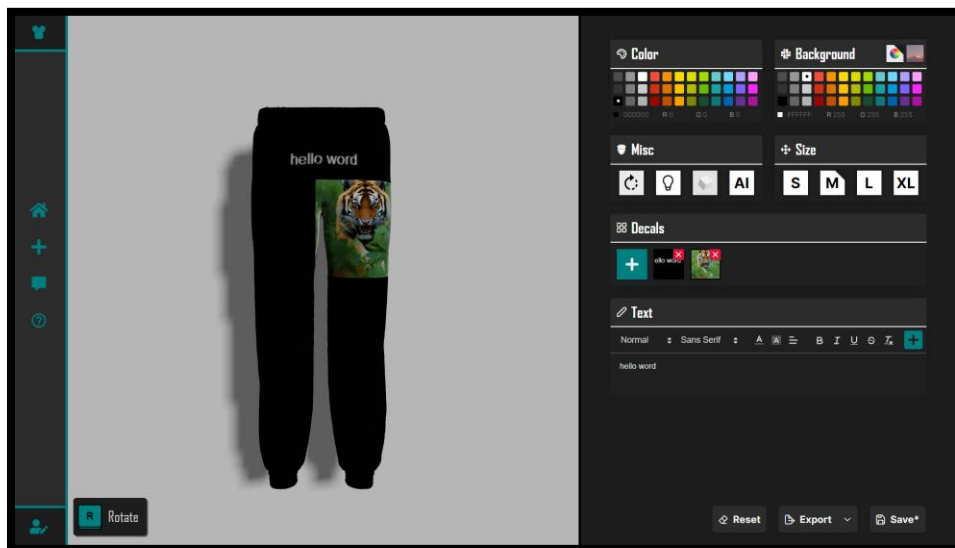
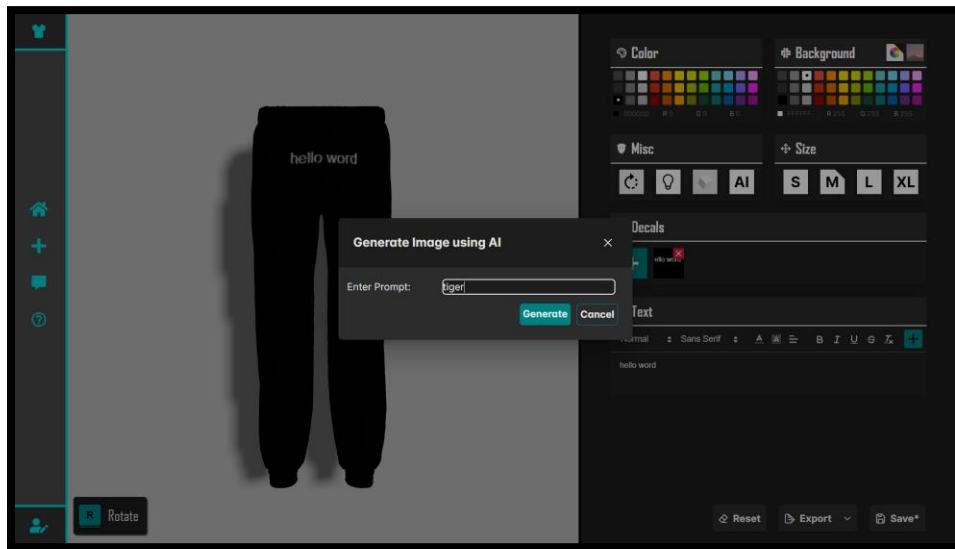


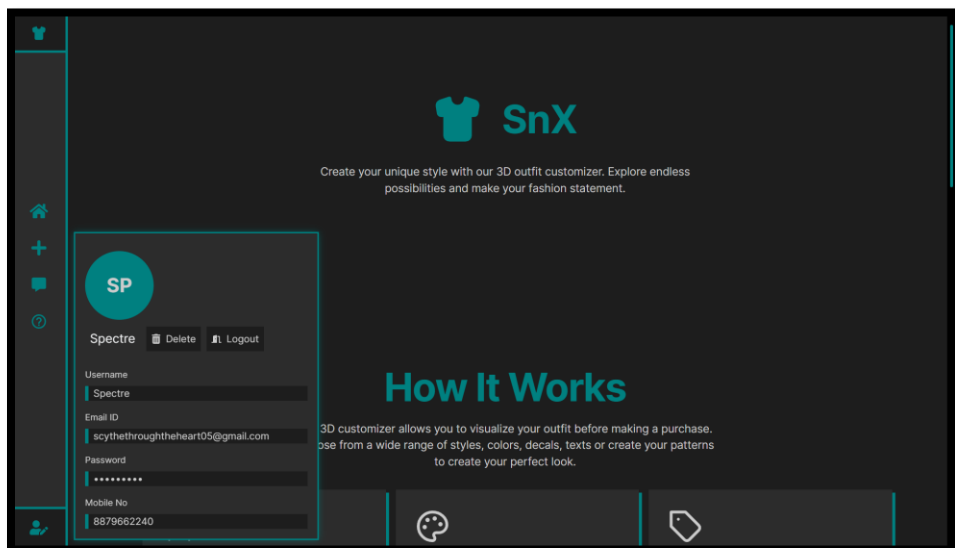
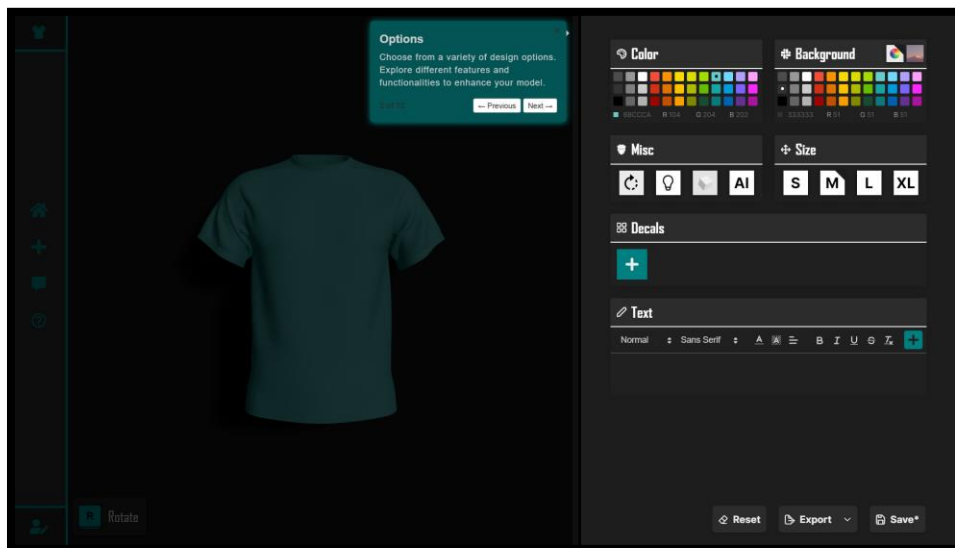
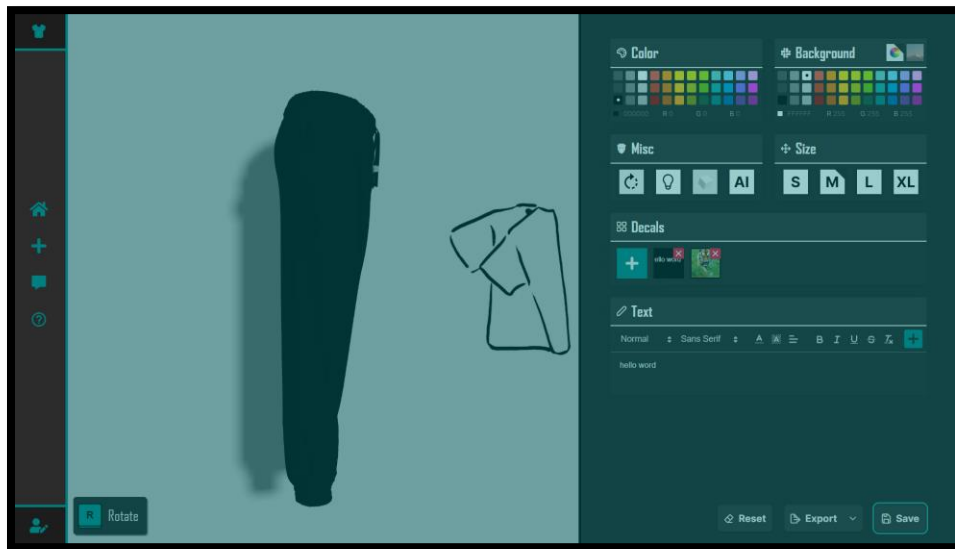












Chapter 8: Future Development and Conclusion

Content:

- 8.1. Introduction
- 8.2. Limitations
- 8.3. Future Enhancement
- 8.4. Reference and Bibliography

8.1. Introduction

The introduction to the future development of the "SnX 3D Outfit Customizer" project provides a concise overview of the current system while delineating the rationale for forthcoming enhancements. It articulates the vision for future upgrades, such as advanced 3D modeling techniques and expanded feature sets, driven by evolving user preferences and technological innovations. Anticipated benefits, including heightened customization options and an enriched user experience, underscore the value proposition of investing in future development endeavors. The introduction culminates with a call to action, urging stakeholders to collaborate in shaping the project's evolution, fostering a sense of collective ownership and shared dedication to driving innovation in personalized fashion technology.

8.2. Limitations

- Limited availability of fabric textures and patterns.
- Compatibility issues with certain devices or browsers.
- Potential privacy and security concerns related to user data.

8.3. Future Enhancement

The future enhancement of the "SnX 3D Outfit Customizer" project involves strategic improvements and expansions aimed at optimizing its functionality, customization options, and user experience. Key aspects of future enhancement may include:

- **Advanced 3D Modeling Techniques:** Implementing cutting-edge 3D modeling techniques and tools to enhance the platform's ability to create lifelike and customizable outfit ensembles with greater precision and detail.
- **Integration with Next-Generation Technologies:** Exploring opportunities to integrate next-generation technologies such as augmented reality (AR), virtual reality (VR), and artificial intelligence (AI) to enrich the user experience and provide immersive and interactive outfit customization features.
- **Enhanced User Interface and Design:** Improving the user interface and design to create a more visually appealing and intuitive experience for users, enabling seamless navigation and interaction with the platform's 3D modeling tools and customization options.
- **Mobile Application Development:** Developing a mobile application for the SnX platform to extend its accessibility and usability to users on-the-go, allowing them to create, customize, and view outfit ensembles from their smartphones or tablets.
- **Collaboration with Fashion Brands and Designers:** Establishing partnerships with fashion brands, designers, and industry influencers to offer exclusive collections, style recommendations, and fashion trends within the SnX platform, providing users with access to curated and premium outfit options.

8.4. Reference and Bibliography

- [1] Dual-Mode User Interfaces for Web- Based Interactive 3D Virtual Environments Using Three.js - Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrallo – 201
- [2] React JS – An Emerging Frontend Javascript Library - Pratik Sharad Maratkar and Pratibha Adkar – 2021
- [3] Frontend Development with React.js - Anjali Rananavare – 2022
- [4] React Apps with Server- Side Rendering: Next.js - Harish AJartarghar, Girish Rao Salanke, Ashok Kumar A.R, Sharvani G.S, Shivakumar Dalali – 2022
- [5] React JS (Open Source JavaScript Library) - Alok Kumar Srivastava, Vaishnavi Laxmi, Payal Singh, Km Pratima, Vibha Kirti – 2022
- [6] Modern Web- Development using React.js - Bhupati Venkat Sai Indla and Yogeshch – 201
- [7] Review on React JS - Dimpy Bansal – 2020
- [8] Role of Node.js in Modern Web Application Development - Ghansham Jadhav¹, Flavia Gonsalves² – 2020
- [9] Comprehensive Analysis of React-Redux Hybrid App Development Framework - Shravan G V, Prof. Anitha Sandeep – 2020
- [10] Survey And Analysis Of Rendering Realtime 3D Object On Cross- Browser & Cross-Platform Using WebGL - Yogiraj Patil, Kirti Wanjale – 2020
- [11] Robust Real-Time Shadows for Dynamic 3D Scenes on the Web - Tim Nicolas Eicke, Yvonne Jung, and Arjan Kuijper – 2014
- [12] Movie Data Visualization Based on WebGL - Min Li, Chunfang Li – 2021
- [13] Comprehensive Analysis of React-Redux Development Framework - Shravan G V and Prof. Anitha Sandeep – 2020
- [14] Server- Based Rendering of Large 3D Scenes for Mobile Devices Using G- Buffer Cube - Mapsn Juergen Doellne, Benjamin Hagedorn – 2020
- [15] The Research and Design Of 3D Web Guide System Based On WebGL - Cui Peng – 2021
- [16] Performance Optimization using MERN stack on Web Application - Sourabh Mahadev Malewade, Archana Ekbot – 2021
- [17] Efficient visualization of 3D models by web browser - Bartosz Sawicki and Bartosz Chaber – 2013
- [18] 3D Rubik's Cube - Online 3D Modeling System Based on WebGL - Buyun Sheng, Feiyu Zhao, Chenglei Zhang, Xiyan Yin, Yao Shu – 2017
- [19] WEBAPP SERVICE FOR BOOKING - Saundariya K, Prabakaran D, - 2021
- [20] HANDYMAN USING MONGO DB, EXPRESS JS, REACT JS, NODE JS - Abirami M, Srimathi B, Senthil Kumaran R, Nagarajan G (IEEE Member) – 2021

- [22] Research and Application of Web3D Exhibition Based on WebGL and Html5 - M.J. Bian, J. Gao, H.H. Gao, J.P. Xu – 2015
- [23] Web 2.0 and Virtual World Technologies: A Growing Impact on IS Education - Albert L. Harris and Alan Rea - 2009
- [24] Investigating Web3D topics on StackOverflow: a preliminary study of WebGL and Three.js - Farag Almansoury, Sègla Kpodjedo, and Ghizlane El Boussaidi – 2020
- [25] Immersive 3D Modeling with Blender and Off-the-Shelf Hardware Matthew Stanton, Thomas Hartley, Fernando Loizides, and Adam Worrall - 2020
- [26] Prasetyo, F. A. (2019). Badan Kepegawaian Negara (BKN). *tribunnewswiki*. <https://www.tribunnewswiki.com/2019/10/24/badankepegawaian-negara-bkn> (accessed Nov. 02, 2021).
- [27] Yogyakarta, K. R. I. B. (2021). Wakil Kepala Bkn: Siasn Solusi Benahi Kualitas Data Kepegawaian. *Yogyakarta.Bkn.Go.Id.* (accessed Nov. 02, 2021).
- [28] Falih, F. (2018). A Review Study of Information Systems. *International Journal of Computer Applications*, 179(18), 15–19. doi: 10.5120/ijca2018916307.
- [29] Venkat, B., Indla, S., Puranik, Y., Student, P. G., & College, P. E. S. M. (2021). Review on React JS. *Journal Title*, 5(4), 1137–1139.
- [30] Bhalla, A., Garg, S., & Singh, P. (2020). Present Day Web-Development Using ReactJS. *International Research Journal of Engineering and Technology*, 7(5), 1154–1157.