Muhammad Ahmad Fraz (332779)

Jamshaid Khalid (352748)

Project Report

Outfit Genius

Table of Contents

[1. Introduction 2](#_Toc135846490)

[2. System Architecture 2](#_Toc135846491)

[2.1 Front-End 2](#_Toc135846492)

[2.2 Back-End 2](#_Toc135846493)

[2.3 Recommendation Engine 2](#_Toc135846494)

[2.4 Distributed Computing Concepts 2](#_Toc135846495)

[3. Functionality 3](#_Toc135846496)

[3.1 Image-Based Recommendation 3](#_Toc135846497)

[3.2 Brand Searching 3](#_Toc135846498)

[4. Implementation Challenges and Solutions 3](#_Toc135846499)

[4.1 Integrating Python Model with Node.js 3](#_Toc135846500)

[4.2 Load Balancing and Fault Tolerance 3](#_Toc135846501)

[5. Results and Evaluation 3](#_Toc135846502)

[4. How to Run the Project? 4](#_Toc135846503)

[Presentation Slides Canva Link 4](#_Toc135846504)

[Conclusion 4](#_Toc135846505)

[Future Enhancements 5](#_Toc135846506)

[References 5](#_Toc135846507)

[Node.js: 5](#_Toc135846508)

[Express.js: 5](#_Toc135846509)

[Http Proxy Package: 5](#_Toc135846510)

[Child Process package: 5](#_Toc135846511)

[ResNet50 (Convolutional Neural Network architecture): 6](#_Toc135846512)

[K-Nearest Neighbors (KNN) algorithm: 6](#_Toc135846513)

[React.js: 6](#_Toc135846514)

[MongoDB: 6](#_Toc135846515)

Distributed Computing Semester Project

Project Report: Outfit Genius

# 1. Introduction

The Outfit Genius project aimed to develop a recommendation system that provides users with outfit and shoe recommendations based on their input image. The project utilized the ResNet50 model and K-Nearest Neighbors (KNN) algorithm to generate accurate and relevant suggestions. Additionally, the project incorporated various technologies, including the MERN stack, distributed computing concepts such as load balancing and fault tolerance, and brand searching functionality. This report provides an overview of the project, its implementation details, and the achieved results.

# 2. System Architecture

The Outfit Genius system employed a client-server architecture, with the following components:

## 2.1 Front-End

The front-end of the application was built using the MERN stack, which consists of MongoDB as the database, Express.js as the web application framework, React.js for the user interface, and Node.js as the back-end runtime environment. The user interface allowed users to upload images and interact with the recommendation system.

## 2.2 Back-End

The back end of the application, implemented in Node.js, played a crucial role in processing user requests, communicating with the machine learning model, and retrieving relevant recommendations. The "child process" package in Node.js facilitated the integration of the Python-based ResNet50 model and KNN algorithm into the Node.js environment.

## 2.3 Recommendation Engine

The recommendation engine was the core component of the Outfit Genius system. It leveraged the power of the ResNet50 model to extract relevant features from the input image and then used the KNN algorithm to find similar images from the dataset. By utilizing machine learning techniques, the system provided accurate and personalized recommendations to the users.

## 2.4 Distributed Computing Concepts

To enhance system performance, the project implemented distributed computing concepts such as load balancing and fault tolerance. The "http-proxy" package was utilized to create multiple endpoints and assign health values to each endpoint. The Round Robin Algorithm ensured load balancing, while health checks enabled fault tolerance by redirecting requests to healthy endpoints.

# 3. Functionality

## 3.1 Image-Based Recommendation

Users could upload images of outfits or shoes through the user interface. The system processed the input image using the ResNet50 model to extract features. The KNN algorithm then matched these features with existing images in the dataset to recommend similar outfit or shoe images to the user. The system provided five recommendations based on the input image.

## 3.2 Brand Searching

The project also implemented brand searching functionality, allowing users to specify the brand, type, and price range of the desired outfit or shoe. The system retrieved relevant items based on the user's input and presented them as search results.

# 4. Implementation Challenges and Solutions

During the development of the Outfit Genius project, several challenges were encountered. Some notable challenges included:

## 4.1 Integrating Python Model with Node.js

Integrating the Python-based ResNet50 model and KNN algorithm with the Node.js environment posed a challenge. However, the "child process" package provided a viable solution by facilitating communication between the two languages.

## 4.2 Load Balancing and Fault Tolerance

Implementing load balancing and fault tolerance required careful consideration of system architecture. The utilization of the "http-proxy" package enabled the creation of multiple endpoints and the implementation of the Round Robin Algorithm, ensuring efficient load distribution. Health checks were implemented to redirect requests to healthy endpoints, ensuring fault tolerance.

# 5. Results and Evaluation

The Outfit Genius project achieved the following results:

* The recommendation system effectively provided five accurate and relevant outfit or shoe recommendations based on the input image.
* The brand searching functionality successfully retrieved and presented search results based on user-specified brand, type, and price range.
* The integration of distributed computing concepts improved the system's performance, enabling efficient load balancing and fault tolerance. The Round Robin Algorithm effectively distributed incoming requests among multiple endpoints, preventing any single endpoint from becoming overwhelmed. The health checks ensured that requests were routed to healthy endpoints, enhancing fault tolerance and overall system reliability.
* User feedback and testing played a vital role in evaluating the system's effectiveness. User satisfaction surveys, feedback forms, and usability testing sessions were conducted to gather insights and make improvements to the system. The feedback received from users helped refine the recommendation algorithm and enhance the brand searching functionality.

# 4. How to Run the Project?

Install Node 17 or any higher version, unzip the project folder and follow the steps below:

1. In outfit-ai-backed folder, type the command:

npm install

2. In outfit-genius folder, type the command:

npm install

3. Open 4 terminals:

In outfit-ai-backend folder, for each terminal, type:

npm run dev

npm run start

npm run load

4. In outfit-genius folder, type:

npm run dev

Now, dependencies are installed, 2 instances of server, 1 server and 1 GUI client is running.

Go to <http://127.0.0.1/5173> on your browser and explore the app.

# Presentation Slides Canva Link

<https://www.canva.com/design/DAFjRjGmlR0/w7z7rn-EnwnYHO7SzOiFJQ/edit?utm_content=DAFjRjGmlR0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

# Conclusion

The Outfit Genius project successfully developed a recommendation system that utilized the ResNet50 model and KNN algorithm to provide accurate and personalized outfit and shoe recommendations based on user-uploaded images. By implementing the MERN stack, the project supported a robust front-end and back-end architecture, facilitating a seamless user experience. The integration of distributed computing concepts, including load balancing and fault tolerance, enhanced system performance and reliability.

Throughout the project, challenges were overcome through the utilization of the "child process" package in Node.js, enabling the integration of the Python-based model, and the "http-proxy" package, facilitating load balancing and fault tolerance.

The Outfit Genius project showcased the potential of machine learning algorithms and distributed computing concepts in the fashion industry. The recommendation system and brand searching functionality provided users with a personalized and efficient shopping experience, improving user satisfaction and engagement.

# Future Enhancements

To further enhance the Outfit Genius system, the following improvements and additions could be considered:

* Integration of additional machine learning models and algorithms to improve the accuracy and diversity of recommendations.
* Incorporation of user feedback mechanisms, such as ratings and reviews, to refine and personalize recommendations based on user preferences.
* Expansion of the dataset to include a wider range of outfits and shoe images, allowing for a more comprehensive recommendation system.
* Implementation of advanced image processing techniques to extract more detailed features from input images, enhancing the precision of recommendations.
* Integration with external APIs and e-commerce platforms to provide real-time availability and pricing information for recommended items.
* Implementation of social sharing features, enabling users to share their favorite outfits and shoe recommendations with friends and followers.
* By incorporating these enhancements, the Outfit Genius system can continue to evolve and deliver an even more compelling and personalized fashion recommendation experience to its users.

# References

## Node.js:

Official website: <https://nodejs.org/>

Documentation: <https://nodejs.org/docs/>

## Express.js:

Official website: <https://expressjs.com/>

Documentation: <https://expressjs.com/en/4x/api.html>

## Http Proxy Package:

npm package page: <https://www.npmjs.com/package/http-proxy>

GitHub repository:<https://github.com/http-party/http-proxy>

## Child Process package:

npm package page: <https://www.npmjs.com/package/child_process>

Documentation: <https://nodejs.org/api/child_process.html>

## ResNet50 (Convolutional Neural Network architecture):

Paper: ***"Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun”***

## K-Nearest Neighbors (KNN) algorithm:

Article: "K-Nearest Neighbors (KNN) Algorithm for Machine Learning" by Jason Brownlee: <https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/>

## React.js:

Official website: <https://reactjs.org/>

Documentation: <https://reactjs.org/docs/getting-started.html>

## MongoDB:

Official website: <https://www.mongodb.com/>

Documentation: https://docs.mongodb.com/

This concludes the project report on Outfit Genius.