Smart AI Mirror

ECE 4333

Interim Project Report

Texas Tech University

Electrical and Computer Engineering

Khisa-Lee Lebrun

R11805898

Group Members: Boyd Paxton

July 2025

Abstract

This paper presents the design and implementation of a smart fashion evaluation system powered by a Raspberry Pi 5 [1], an AI HAT equipped with a Hailo AI accelerator [2], and a custom-built fashion analysis algorithm. The system integrates computer vision and machine learning models within a responsive web application to detect and evaluate outfits in real time. Both hardware and software elements are detailed, from YOLO-based [3] image detection to outfit grading based on body shape, color coordination, and clothing combinations. Project-specific requirements are broken down and assessed to ensure that performance, responsiveness, and user interactivity targets are met. The document concludes by analyzing final results, providing insight into the system's performance, its practical applications in personalized style recommendation, and the approach taken to embed AI capabilities in a low-power, edge-computing platform.

Acknowledgement

The results shown in the following report would not have been possible without the help and guidance of classmates and faculty who provided mentorship and advice, as well as the friends who provided moral support. Special recognition is extended to all ECE professors and teaching assistants who always took the time to point the project in the right direction and lend out advice while issues were faced along the making of this project. It is crucial to note that without all this assistance, the project would not be where it is currently.

Table of Contents

Table of Figures

I. Introduction

As routine as choosing an outfit may seem, the integration of artificial intelligence into daily fashion decisions remains largely untapped, especially in accessible, real-time, and personalized formats. With growing interest in self-expression, sustainability, and digital health, the development and testing of AI-based fashion tools is becoming increasingly relevant. This project focuses on the design and implementation of an intelligent fashion analysis system built on a Raspberry Pi 5, paired with an AI HAT featuring a Hailo accelerator. The system connects to a dynamic web-based user interface to display detailed outfit evaluations and personalized feedback. However, the sophistication of such a system requires more than visual detection alone. It must ensure efficient performance, user privacy, and data security, particularly when handling personal images and body measurements. These concerns are addressed through optimized algorithms and local edge processing, which minimizes the need for cloud interaction and helps protect user data.

The objective of this project is to deliver a complete AI-powered smart mirror solution that integrates hardware components such as cameras and a compact display with software modules for body-shape classification, color coordination analysis, and outfit grading. The final system is designed to provide users with practical, fashion-forward insights while maintaining responsiveness, privacy, and ease of use. This project is scheduled for final presentation on August 6, 2025, as detailed in Appendix A.

II. Body of Technical Report

1. Hardware

A. Raspberry Pi 5

As mentioned, a Raspberry Pi 5 was used for this project. Its compact size, significantly enhanced CPU and GPU performance, and expanded memory options make it well-suited for edge computing applications. The built-in support for dual 4K HDMI outputs, high-speed USB 3.0 ports, and improved thermal management allows for a smoother user experience and greater reliability in continuous operation. Additionally, its integrated Wi-Fi and Bluetooth capabilities, along with a strong open-source community, make it an ideal choice for developing and deploying interactive, AI-enhanced systems such as a smart fashion mirror. These features enable the Raspberry Pi 5 to serve as a powerful and cost-effective base for real-time image processing and user interface control.

B. AI Hat

To enhance the system's processing capabilities for deep learning tasks, the Raspberry Pi 5 is paired with an AI HAT equipped with a Hailo AI accelerator. The Hailo chip is specifically designed for efficient edge inference, delivering high-performance neural network processing with low power consumption. This hardware component enables the system to run complex computer vision models, such as YOLO-based outfit detection and classification algorithms, directly on the device without relying on cloud resources. By offloading AI workloads to the Hailo accelerator, the AI HAT ensures smooth and real-time performance while freeing up the Raspberry Pi's CPU for other tasks such as user interface control and data management. This setup provides the necessary hardware foundation for a responsive and privacy-conscious smart fashion assistant.

C. Pi Camera

To enable outfit detection, a Raspberry Pi Camera [4] is used and connected to the Raspberry Pi 5 via a flat ribbon cable. This camera serves as the system's primary visual input, capturing real-time images of the user for analysis. The use of the dedicated CSI (Camera Serial Interface) port ensures high-quality image transmission with minimal latency, which is essential for running computer vision algorithms effectively. Its compact size and direct integration make it an ideal choice for embedded applications like the Smart Mirror, allowing for seamless image capture without the bulk or complexity of external webcams.

D. Set up

The installation is based on a monitor screen with its feet removed, placed directly against a two-way mirror [5]. The two elements are held firmly together by a custom-designed wooden frame as seen below in figure 1.

Figure 1: Wooden Frame. [6]

This frame provides a slight extra depth to conceal the cables, Raspberry Pi 5, AI HAT, and other electronic components required for the system to function properly. To ensure optimal image and video quality, the Raspberry Pi Camera is positioned above the setup rather than behind the two-way mirror. This placement avoids glare and distortion from the reflective surface, allowing the camera to capture clearer, more accurate visuals for outfit detection. The overall structure provides stability, aesthetic appeal, and protection for the internal hardware, while allowing the user's reflection and the screen display to remain visible when the system is powered on.

2. Web Application

A. Mirror GUI

In order to display multiple pieces of information on the mirror, more precisely on the monitor placed behind the semi-transparent mirror, the open-source platform MagicMirror[7] was used. MagicMirror is a highly flexible and modular smart mirror framework developed specifically for the Raspberry Pi. It comes with a variety of pre-built modules such as clock, weather, calendar, news feeds, and system performance, which can be easily enabled, disabled, or rearranged. The interface is built using web technologies, making it simple to modify through HTML, CSS, and JavaScript. MagicMirror also supports third-party modules and allows developers to create their own plugins based on specific needs as seen below in figure 2.

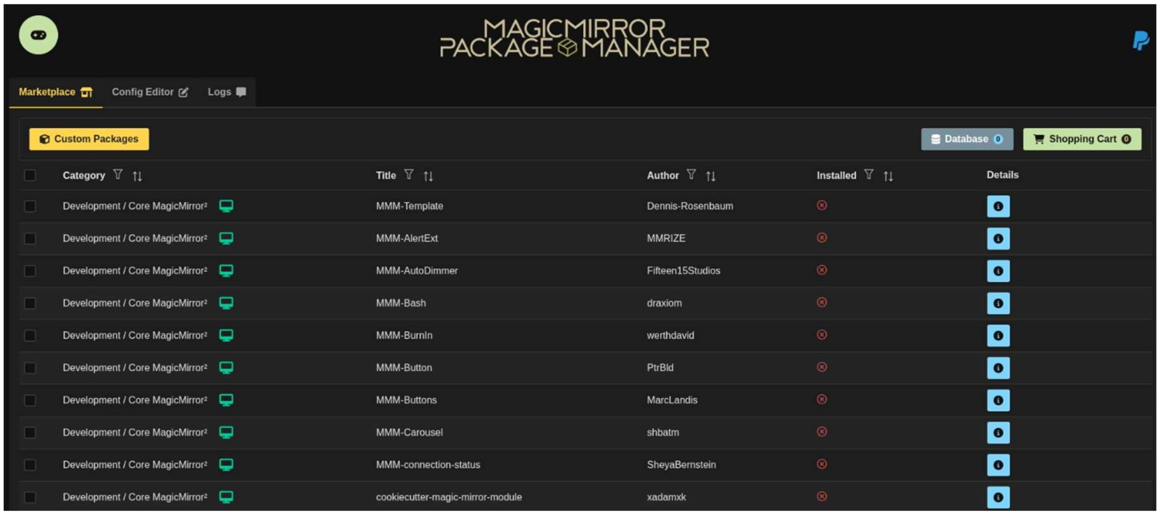


Figure 2: Magic Mirror Template.

For this project, a custom module was created to integrate the additional AI fashion functionality within the existing MagicMirror interface. The framework's lightweight design and strong community support make it an ideal choice for smart mirror projects that require both stability and customization.

B. Fashion AI

i. Dataset

To ensure accurate and relevant outfit detection, a custom dataset of 300 annotated images was created specifically for this project. Each image was carefully labeled using Roboflow [], with clothing items categorized by type, such as tops, bottoms, dresses, shoes, and accessories as seen below in figure 3.

2pt bag	1
2pt coat	1
2pt necklace	1
2pt short	1
sunglasses	1
1pt beanie	2
1pt skirt	2
1pt sport top	2
1ptscarf	2
2pt shirt	2
1pt sport pants	3
2pt belt	3
2pt romper	3

Figure 3: Clothing Items in the Dataset.

These categories allowed for structured training and improved the model's ability to differentiate between specific garment classes. A point-based labeling system was also applied to help identify key visual features and improve detection consistency across varied lighting, angles, and backgrounds. Once fully annotated, the dataset was uploaded to Roboflow and processed to create a high-quality training set as seen below in Figure 4 and Appendix B.

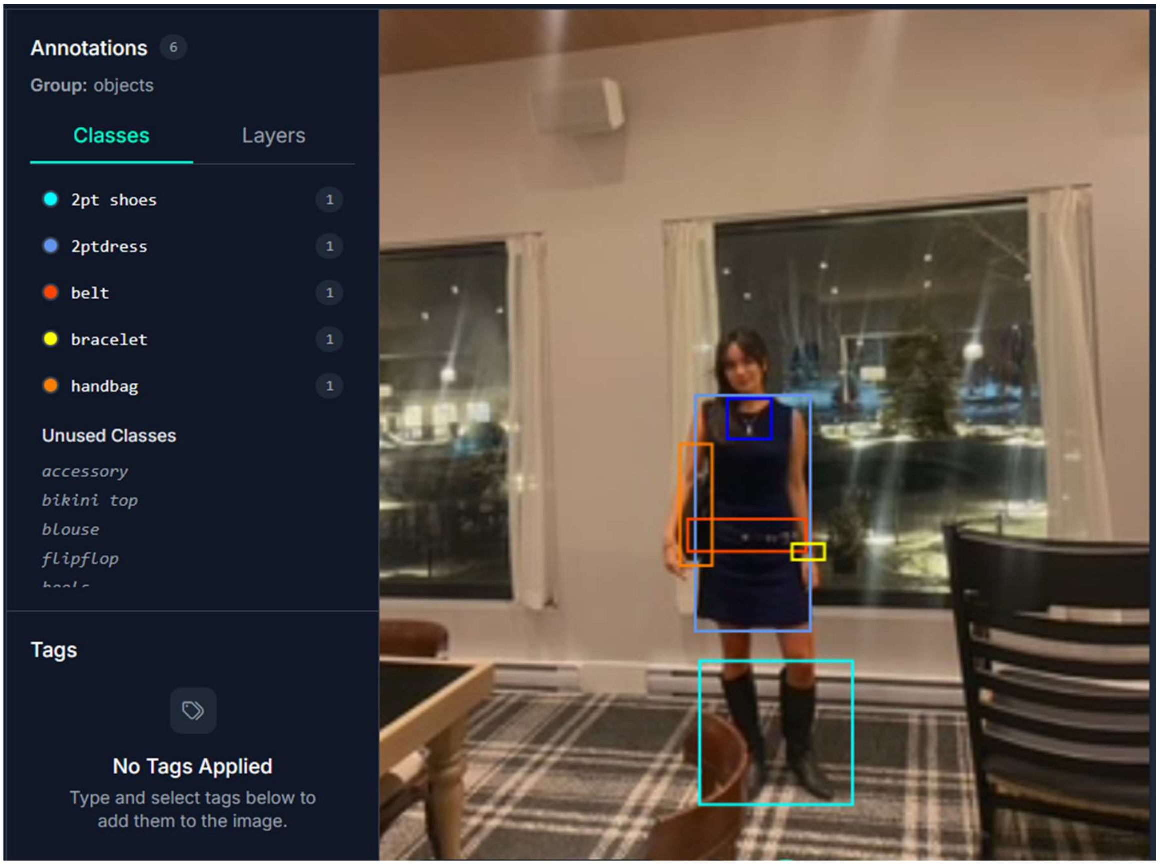


Figure 4: Clothing Detection with Bounding Boxes.

The platform handled image preprocessing tasks such as resizing, augmentation, and annotation format conversion, generating a clean and consistent dataset. This dataset was then used to train a custom object detection model that serves as the foundation for the image recognition tasks within the system.

ii. Algorithm

The software basis for this project is based on Python language. Python is renowned for its extensive library of modules and its wide compatibility with a variety of devices. In this project, Python was used to implement an algorithm that performs outfit evaluation through object detection, color extraction, and body shape classification. The YOLO model is used to detect visible clothing items in a user-captured image. Then a pre-existing YOLO model is used to detect the color of the clothing items. Simultaneously, the user provides body measurements including shoulder width, waist, hips, armpit-to-waist length, and waist-to-hips length. Based on these values, the algorithm determines the user's body shape by evaluating key proportional ratios. After detecting the clothing items and analyzing the color composition, the system applies a point-based grading method. It generates a score, seven points for the outfit, and two possible bonus points for the color combination, as seen in Appendix C. Along with the grading, qualitative feedback and suggestions for improvement are given on the outfit's composition. All processing occurs locally on the Raspberry Pi 5, ensuring low latency and full control over user data. The result is a structured output that includes clothing item detection, body shape classification, color analysis, and a style rating presented through the user interface.

iii. GUI

To display the results of the AI analysis, a custom graphical user interface (GUI) was developed using Python as the core framework. The Eel library [9] was then integrated to enhance the front-end design using standard web technologies including JavaScript, HTML, and CSS. This approach allows for a more modern and user-friendly interface while maintaining seamless communication with the Python backend. The GUI first prompts the user to enter their body measurements, as illustrated in Figure 5.

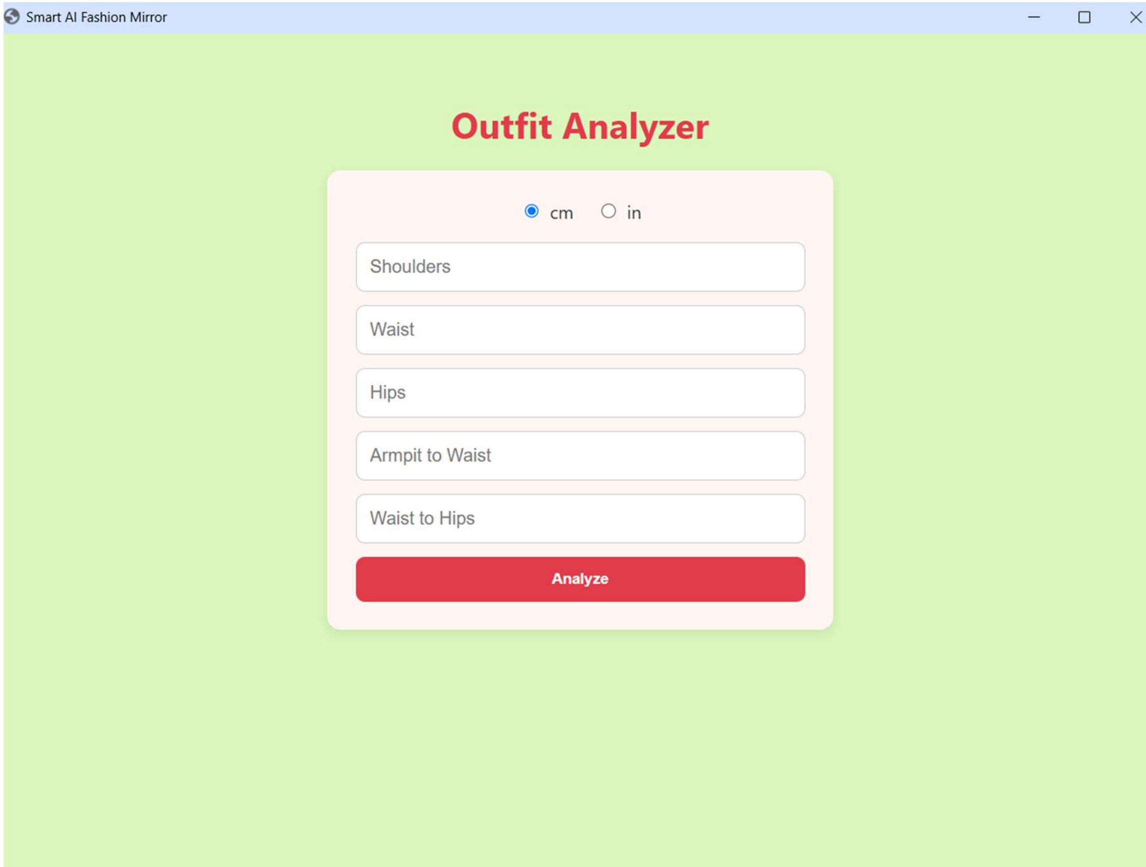


Figure 5: Fashion AI GUI.

Once the measurements are submitted, a short countdown begins to give the user time to position themselves in front of the camera. After the countdown, the system captures an image and proceeds to analyze the user's clothing using the trained detection model. The results are then displayed, including an outfit score and personalized feedback. In cases where the camera may not detect smaller items such as earrings or accessories due to size, lighting, or positioning, the user has the option to manually input any missing elements to ensure a more complete and accurate evaluation as seen below in figure 6.

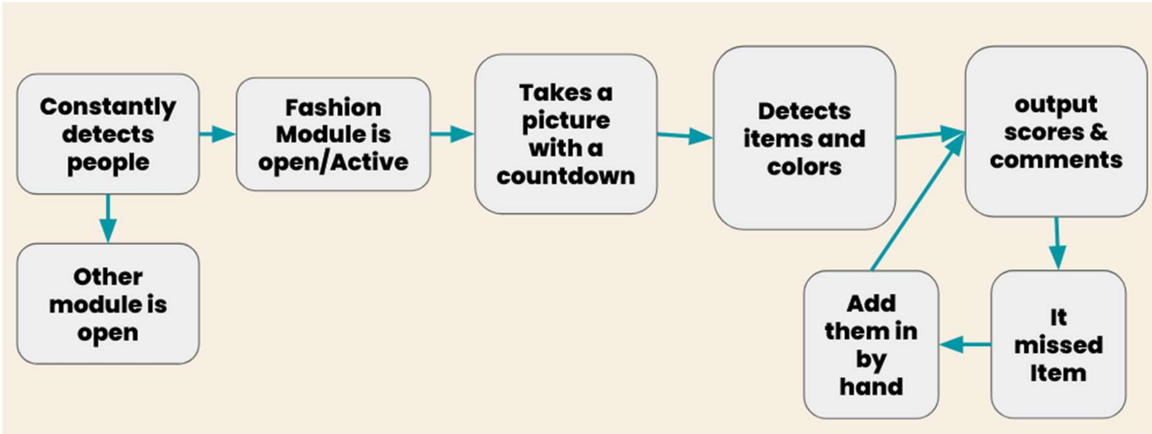


Figure 6: Fashion AI Flowchart within the MagicMirror.

III. Engineering Standards, Specifications, and Intellectual Property Considerations

In this Smart Mirror Fashion AI project, several engineering standards and software development practices were followed to ensure system quality, maintain compatibility, and respect intellectual property guidelines. The project combines hardware and software components, each requiring adherence to specific norms. The frontend interface is built using HTML, CSS, and java through Eel, providing a clean and responsive user experience suitable for real-time outfit analysis and feedback. On the hardware side, integration practices consistent with Raspberry Pi HAT specifications were followed, particularly for interfacing with the Hailo AI accelerator and peripheral components like the camera and monitor. All third-party tools, including machine learning libraries and computer vision models (such as YOLO), were used under appropriate open-source licenses, and proper credit is given where applicable to avoid copyright or intellectual property infringement. Special care was taken to ensure the originality of the outfit grading algorithm and body-shape classification logic, which were custom-developed for this project.

IV. Safety, Public Health, and Welfare Considerations

In this Smart Mirror Fashion AI project, user safety, privacy, and overall well-being are central to the design. The system is engineered to process all image data locally on the Raspberry Pi 5, avoiding the need to transmit personal photos or measurements to external servers. This approach significantly reduces the risk of data breaches and ensures that user information remains confidential. Additionally, the AI HAT and processing units are installed within a ventilated wooden frame that prevents overheating and shields users from exposed wiring or electronic components. To support public welfare and mental health, the fashion analysis algorithm is designed to be constructive and non-judgmental, avoiding negative or harmful feedback. The system offers suggestions that promote self-expression and confidence, rather than criticism. The interface is also optimized for clarity, comfort, and energy efficiency to reduce visual strain and device wear over time. All collected data is securely stored for temporary use only and is either anonymized or deleted after its intended purpose is fulfilled, ensuring users' digital safety and peace of mind. Together, these considerations aim to create a safe, respectful, and health-conscious experience for every user interacting with the smart mirror system.

V. Global, Cultural, Social, Environmental, and Economic Factor Considerations

In this Capstone Laboratory project at Texas Tech University, a strong emphasis is placed on sustainability, efficiency, and responsible engineering. The integration of power management strategies and localized AI processing helps reduce energy consumption while ensuring stable performance. By limiting the need for external servers and minimizing data transmission, the system avoids unnecessary resource usage and reduces the risk of overheating. These measures contribute to extending the lifespan of the hardware and reducing electronic waste. Particular attention is given to protecting key hardware components, including the Raspberry Pi 5 and the AI HAT, which represent the most valuable parts of the system. Together, these components total an investment of 160.76 USD, as detailed in Appendix D. Preserving their functionality through thoughtful design and thermal protection is essential to ensuring both the durability of the system and the conservation of financial and material resources. This approach reflects a strong commitment to sustainable development and ethical engineering practices throughout the project lifecycle.

VI. Conclusion

In conclusion, the development of this Fashion AI Smart Mirror demonstrates the successful integration of multiple hardware and software components, each contributing to the system's overall functionality, efficiency, and user experience. The combination of real-time image processing, personalized fashion analysis, and a responsive graphical interface showcases the potential of artificial intelligence when applied at the edge. The use of local processing through the Raspberry Pi 5 and the AI HAT ensures both privacy and performance, while the modular design powered by MagicMirror allows for a clean and adaptable user interface. This project reflects the growing possibilities of blending AI-driven insights with embedded systems to create intelligent, user-focused applications. As smart technologies continue to evolve, this work highlights how thoughtful engineering and innovation can enhance everyday tools and personal experiences.

VII. References

[1] Raspberry Pi 5 - https://www.raspberrypi.com/products/raspberry-pi-5/

[2] AI Hat with Hailo - https://www.raspberrypi.com/products/ai-hat/

[3] YOLO - https://docs.ultralytics.com/fr/

[4] Pi Camera - https://www.digikey.com/en/products/detail/raspberry-pi/SC1174/24627137?gclsrc=aw.ds&gad_source=1&gad_campaignid=20243136172&gclid=Cj0KCQjwvajDBhCNARIsAEE29WohMc-WCu6M-Bb1LoT_gf6A_pqF3YKRTZveSTrvgOulldthBispF1MaAjx8EALw_wcB

[5] Two-way Mirror - https://www.digikey.com/en/products/detail/raspberry-pi/SC1174/24627137?gclsrc=aw.ds&gad_source=1&gad_campaignid=20243136172&gclid=Cj0KCQjwvajDBhCNARIsAEE29WohMc-WCu6M-Bb1LoT_gf6A_pqF3YKRTZveSTrvgOulldthBispF1MaAjx8EALw_wcB

[6] Wood frame made by Boyd Paxton’s Grand-father.

[7] MagicMirror - https://magicmirror.builders/

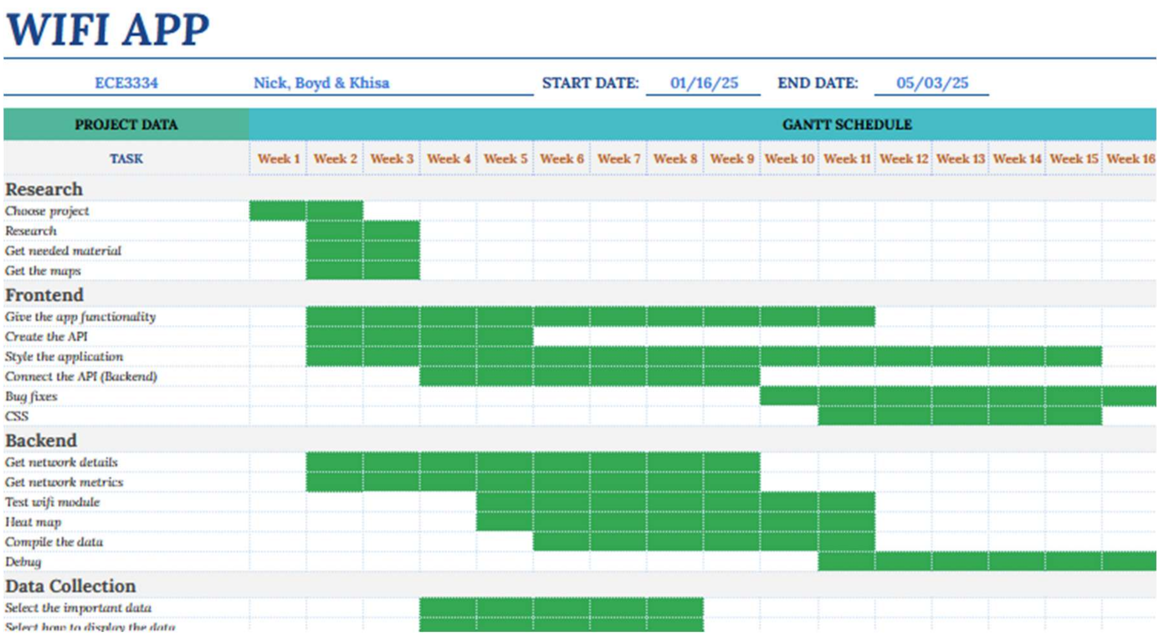
[8] Roboflow - https://roboflow.com/

[9] Eel - https://github.com/python-eel/Eel

VIII. Appendices

Appendix A: Project Gantt Chart.

This section provides a view of the Gantt chart of the project.



This section shows the Gantt chart for this project. The project began on January 16, 2025 and will end on May 3, 2025. The schedule is divided into five main categories: research, frontend development, backend development, data collection and testing. Each task is represented by a green bar indicating its expected duration. Tasks include application creation, API integration, user interface design, network data processing and results analysis. This planning allows you to monitor the project's progress in a clear and organized way, while efficiently distributing responsibilities between group members. The poster also shows the coordination between the various phases, in particular the interconnection between the frontend and backend, essential to the smooth running of the application.

Appendix B: Clothing Items Detection with Bounding Boxes and Accuracy.

This section provides a view of the object detection for the algorithm.

This section illustrates the object detection capability of the algorithm used in the clothing recognition module. The figure demonstrates how the AI model identifies and labels various clothing items on users using bounding boxes and confidence scores. Each detected item is enclosed in a color-coded box, annotated with its corresponding label, like “maxi skirt,” “blouse,” “necklace”, and the confidence level expressed as a percentage. This probability indicates the model's certainty about each detection. For instance, garments like skirts and tops are detected with confidence levels exceeding 98%, showcasing the model's accuracy and reliability. These results validate the effectiveness of the underlying YOLOv8-based detection system integrated with the fashion grading algorithm. The bounding boxes provide a visual reference for how the model interprets and segments fashion items, forming the basis for the outfit scoring and recommendation logic implemented in the project.

Appendix C: Fashion Algorithm Code Output

This section provides a view of the fashion algorithm output.

```
PS D:\SmartAIMirror> python grading.py
Enter shoulder measurement (in inches or cm): 70
Enter waist measurement (in inches or cm): 60
Enter hip measurement (in inches or cm): 62
Score: 6/7 style points + 2/2 color points
Grade: Good
Style feedback: You're almost there. Add one or two accessories or layers to complete your look.
Style suggestions:
  - Consider adding a watch, bracelet, or belt.
  - Try layering with a jacket or overshirt.
Color feedback: Nice balance of neutrals and accent color, it adds personality.
Color suggestions:
Body shape: Inverted Triangle
```

This section shows the output produced by the fashion algorithm based on user measurements and detected clothing items. The output is displayed as a terminal-style response from the grading.py script, which processes shoulder, waist, and hip measurements to evaluate outfit compatibility. The system assigns a score, including points for style, composition and color coordination. In the example, the user receives 6 out of 7 style points and 2 out of 2 color points, resulting in a total score of 8. The grade is marked as "Good." The algorithm also generates style feedback and improvement suggestions, such as adding accessories or layering garments. Color analysis is included to assess the balance between neutral and accent tones. Finally, the algorithm determines the user's body shape based on measurement ratios, providing additional personalized insight.

Appendix D: Material Cost Budget

This section provides the budget for the project.

AI Mirror						
	Running Total			Total Estimate		
Direct Labor:	rate per hour	hrs	total	rate per hour	hrs	total
Boyd	18	50	$900,00	15	200	3 000,00 $
Khisa	18	51	$918,00	15	200	3 000,00 $
DL subtotal:			$1 818,00			6 000,00 $
Material Cost:						
Bought			Price	Budget : $100		
Raspberri pi 5 + AI hat			$160,00	Ai kit raspberry pi 5		160,76 $
pi camera			$26,00	pi camera		26,00 $
two way mirror			$34,00	hdmi cable		8,99 $
				two way mirror		50,00 $
MC subtotal:			$220,00			245,75 $
Total			$2 038,00			6 245,75 $

In summary, this diagram describes the price of each element purchased for the project as well as the cost per hour of labor.