A SOPHISTICATED MOBILE APPLICATION TO DETERMINE HAIRSTYLE AND LOCATE BARBERS USING ARTIFICIAL INTELLIGENCE MODELS AND WEB SCRAPING

Michael Zhang [1], Rodrigo Onate [2]

[1] Oakton High School, 2900 Sutton Rd, Vienna, VA 22181
[2] California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Many people struggle to find personalized hairstyle ideas and are seeking trusted barbers in their area. To solve this, we propose an AI-powered barber app that generates hairstyle images based on user preferences, showcases trending/celebrity styles, and connects users with nearby barbers with trusted reviews and more [7]. One of the major challenges was gathering up to date barber information which was solved using web scraping to automatically collect data such as reviews, hours, and locations. During experimentation, the application was put through prompt engineering experimentation, and zip code testing [8]. Results showed that overly specific prompts led to less accurate hairstyle images, with a surprising drop in quality despite higher detail. Meanwhile, zip code testing revealed that only 1.5% were missing, confirming a highly reliable barber database. These results helped identify blind spots and reinforced the importance of prompt balance and verified data sources. StyleSync is an idea that ultimately is something that people will use because the haircut options are limitless, and people are constantly looking for new styles.

KEYWORDS

Web Scraping, Prompt Engineering, Barber, Hair Style

1. INTRODUCTION

The problem this project aims to solve is the difficulty customers face in finding local barbers, and the difficulty finding the perfect hairstyle that suits your facial features, hair type, and personal style. When finding a reliable barber who you trust, many people rely on word-of-mouth recommendations or trial-and-error, which often leads to unsatisfactory experiences. A haircut in this society can mean many different things - for some, it's about confidence, self-expression, and saving time. A hairstyle that complements an individual's features and preferences can significantly boost self-esteem and confidence [9]. According to a 2021 survey by YouGov, 89% of people believe their appearance affects their confidence levels [1]. In addition, barbers often struggle to attract and retain clients due to a lack of exposure. This poses an economic problem as the barbers are there, just not being seen by customers and therefore creating deadweight loss. Furthermore, on economics, the projected barber job growth rate is 11% from 2018-2028 according to Zippia [2]. This means that more barbers will need more exposure in order to connect to customers. Another reason this problem is important is that the majority of people live busy lifestyles that make it difficult for them to spend time researching hairstyles or finding the

David C. Wyld et al. (Eds): NLCAI, AIFU, CCSEA, BIoT, SEA, SIPRO, BDML, CLOUD – 2025
pp. 281-291, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151021

right barber. In the long run, this problem affects both barbers and customers because barbers won't be able to maximize profits while customers will be wasting time researching a haircut. A new survey from YouGov Omnibus finds that only about a third of Americans (38%) always see the same person for a haircut [3]. This leaves the other 62% of Americans constantly looking for a new barber.

StyleSage personalized hairstyle recommendations using facial recognition and face shape classification to tailor suggestions to each user. This method is effective in creating highly individualized results. However, it raises privacy concerns, as uploading facial data can make users feel exposed or uncomfortable. It also relies on accurate facial analysis, which may not always make features correctly. StyleSync improves on this by letting users input personal details manually. This protects user privacy while still providing tailored results, making it more secure and user-controlled.

Web scraping is a powerful way to get up-to-date information, such as barber addresses, ratings, and services provided, from web listings. While it efficiently collects vast amounts of data, its biggest problem is that it might pick up unwanted information. StyleSync eliminates this issue by cleaning and filtering the scraped data, offering only cleaned-up, easy to read, and user-accessible information in the app interface.

The method described in “Recommendation System for Hairstyle Based on Face Recognition Using AI and Machine Learning” uses facial features to recommend hairstyles. It’s accurate in theory but has practical flaws, especially when facial recognition misinterprets features or functions inconsistently across diverse users. StyleSync bypasses this by collecting data through user input rather than facial scans. This helps eliminate errors related to misidentification and ensures that recommendations are based on clear, self-provided traits, improving accuracy while also avoiding ethical and privacy concerns.

The solution to this problem is to build a barber app that uses AI-driven image generation that fits user preferences to recommend personalized hairstyles, updating trending and celebrity styles, and connecting users to nearby barbers for booking and consultation. This app bridges the gap between providing trending and celebrity styles while also leveraging AI to recommend personalized styles, and connecting users with nearby barbers. This is an effective solution because the AI is able to tailor hairstyles to the individual users, ensuring higher satisfaction and engagement. By giving the user an option to input personalized information, the AI model would be able to create a desired hairstyle that fits the user's input. For example, if a user has straight hair texture, the generative AI would produce a straight hair texture hairstyle [10]. No matter if the user isn’t satisfied with a style through the AI model, the trending and celebrity styles would always be available. Furthermore, by connecting users to barbers and enabling booking, the app simplifies the entire process, saving time and effort. Users can view barber profiles, portfolios, ratings, and reviews to make informed decisions making it even more effective. Many existing apps rely on set recommendations or user-driven searches, which can be overwhelming and impersonal. This solution uses AI personalized suggestions, making it more individualized and effective. Moreover, social media inspirations like Instagram or TikTok also don’t provide personalized recommendations or barber connections. Therefore, this AI powered app is the best solution to this problem.

The first experiment focused on the personalized image generation screen, which uses the DALLE 3 model. It tested how changes in the AI prompt including hairline, texture, and length affected the expected image accuracy. This is called prompt engineering where changes to the prompt are taken as data until the best prompt to produce the best match is found. The most surprising finding was that the most specific prompt didn’t produce the most accurate image

which is a common misconception. The second experiment tested the app's zip code database by comparing it to an official USPS list to identify missing zip codes. Out of 200 zip codes tested, only 3 were missing, indicating a 1.5% error rate. This low percentage suggests the database is reliable. However, if the original dataset wasn't regularly updated or verified, more missing zip codes would be created as the universal zip code dataset is always changing.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. How to Gather the Information

A problem I could face when building the app is how to gather all the barber information. I need to figure out how to gather the information of each individual barber to display on my app. I would have to include all reviews, hours, and maybe even location. This would be very difficult to do by hand and if done manually, the barber directory of the app wouldn't always be up to date. Therefore, it is important to come up with a solution to automatically gather this information and display it on the app. I could resolve this problem by using web scraping to take information on barbers and then being able to manipulate it.

2.2. The User Interface

Another problem I could face is problems in the user interface or UI. Problems in the user interface would cause users to struggle to navigate the app or find the information they need. For example, if the layout is cluttered or buttons are too small, it could frustrate users and discourage them from using the app. To resolve this, I could have many people test it out before publishing the app to mitigate as many bugs in the UI as possible. It would also be important and beneficial to ask testers for feedback on the user-friendliness of the app and not just errors.

2.3. Prompt Engineering

Furthermore, I could also face problems with prompt engineering. Prompt engineering is where you design the prompt or directions to feed into the AI model. An important prompt is very important to get what you need as the AI model generates based on the prompt. If a prompt is vague or poorly structured, the model might generate incomplete or wrong information. This could lead to unreliable app functionality, especially if users are expecting precise results. To resolve this, I can repeatedly test different prompts until I get the best one with the most accurate results. This would require me to analyze patterns and refine the prompt a lot.

3. SOLUTION

The main structure of my program is the connection of the local barber search screen, personalized styles screen, and profile screen. These major components make up the main structure of the app and guide the user journey through the flow of the app. The app begins with a splash screen that leads to the home screen, where users can open the drawer and choose to search for local barbers, generate a personal style, or create a profile. The local barbers search feature allows users to search for barbers near them and browse a directory of barbers, view individual profiles, check reviews, operating hours, and locations. bThis information could be gathered and updated automatically through web scraping, ensuring accuracy and real-time updates. The personalized styles screen allows for user input where a specifically engineered prompt combines with user input to allow the AI model to generate potential styles for the user

[11]. The profile management system ties the app together, allowing users to create accounts, log in, and access a personalized dashboard. The login and signup are split into two different screens for easier user navigation. Once a user logs in, they get directed to the profile display screen to display their information. This also happens after a user signs up. Throughout the entire app, the drawer will be available in the top left corner to let users navigate the app wherever they please. Furthermore, they would also have the luxury of a back button to return to a previous screen. All screens and aspects of the app creates a dynamic and user-centered experience.

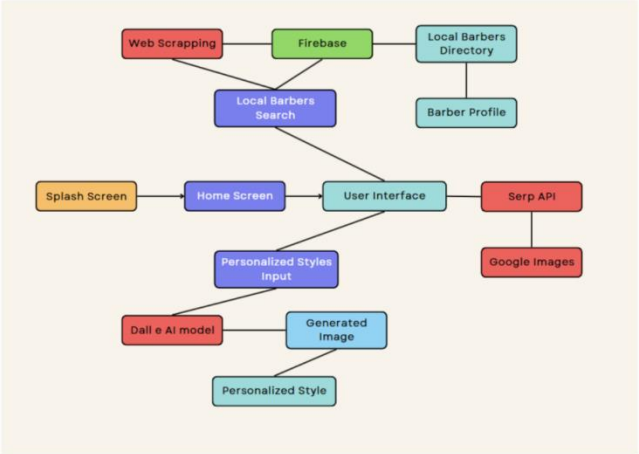

Figure 1. Overview of the solution

The Local Barbers Search helps users find nearby barbers with details like reviews, hours, and locations. It uses web scraping to gather real-time data and therefore keeping the directory updated. One of the services used to implement this system includes Amazon Web Services to temporarily store the information that was gathered.

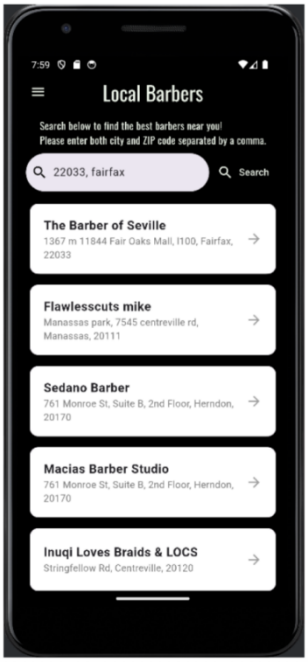

Figure 2. Screenshot of local barbers

```
Future<void> _searchBarbershops() async {
  final String input = _searchBarController.text;
  final List<String> parts = input.split(',');

  if (parts.length < 2) {
    print('Please enter both city and ZIP code seperated by a comma.');
    // Show an alert dialog to the user
    _showAlertDialog(context);
    return;
  }

  final String city = parts[0].trim();
  final String zipcode = parts[1].trim();
  final String url =
      'http://184.72.20.54:5000/get_barbershops/$city/$zipcode';

  print('Request URL: $url');

  setState(() {
    _loading = true;
  });
```

Figure 3. Screenshot of code 1

The process _searchBarbershops() starts by retrieving the text input from _searchBarController() and storing it in the variable “input”. The function then breaks the input into parts using the comma with one part expecting the city and the other expecting the ZIP code. If fewer than two parts are provided, it prompts an alert dialog and returns. The city and ZIP code are stored in variables named city and zipcode, respectively. If the input is valid, the code extracts the city and ZIP code, trims any excess spaces, and creates a URL. The URL contains the information from the variables city and zipcode. This URL is sent to a backend server where it searches for data on barbers and sends it back. In the process of gathering data on barbers, the information is cleaned up as unnecessary spaces, characters, and punctuation are removed. This code is run when the function _searchBarbershops() is called.

The celebrity and trending styles inside the home page allows for users to discover new hairstyles based on celebrities and styles that are currently trending. It uses Serp API to access and grab google images to display it on the home screen [12]. The UI on this function also includes many flutter functions for optimal user convenience such as scrollable images.

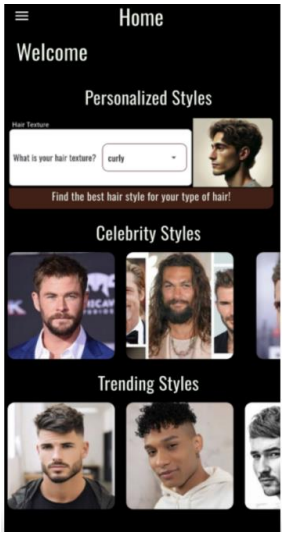

Figure 4. Screenshot of home page

```
Future<List<String>> fetchCelebrities() async {
  const String apiKey =
      "1b179a642c933a570453a3b12ee812184c1398fa8ad454df3e847855105bbe43";
  const String query = "men celebrity hairstyle";
  const String location = "San Francisco, CA, California, United States";

  final Uri url = Uri.parse(
    "https://serpapi.com/search.json?engine=google_images&q=$query&location=$location&api_key=$apiKey",
  );

  try {
    final response = await http.get(url);

    if (response.statusCode == 200) {
      final Map<String, dynamic> data = jsonDecode(response.body);
      final List<dynamic> images = data['images_results'] ?? [];

      // Extract URLs from the response
      return images
          .map<String>((image) => image['thumbnail'] as String)
          .toList();
    } else {
      print("Failed to fetch images: ${response.statusCode}");
      return [];
    }
  } catch (e) {
    print("Error: $e");
    return [];
  }
}
```

Figure 5. Screenshot of code 2

This code is a function called fetchCelebrities(), and it helps get pictures of celebrities' hairstyles from the internet. It uses Serp API to ask a server for images. First, the code creates variables like the apiKey to allow access to the API; the query or the search term; and the location or where the search is happening. Then, it creates a url, which is the address where the request is sent. The try block sends a request using http.get(url). If the server responds with statusCode 200, it works and the response is turned into a list of images. If the response code is not 200, then something went wrong and "Failed to fetch images" is printed along with the statusCode 200. This prevents unwanted things to be shown in case of a malfunction in the app. This function runs when the program wants to get celebrity hairstyle images, and the server's job is to find the images and send them back.

The DallE3 AI model is used to generate a personalized style based on the users' hairline, desired length, and hair texture. It uses the AI model to generate images of this personalized hairstyle from all angles including good lighting to emphasize the actual hairstyle. The main component to allowing this function to work is the dall e3 AI model and its image generation capabilities.

Figure 6. Screenshot of hairstyle

```
Future<void> _generateImage() async {
  final prompt =
  '''
       Generate three images of a male with the following characteristics:
       Hair Texture: ${widget.hairTexture}
       Hair Length on top: ${widget.hairLengthTop}
       Hair Length on side: ${widget.hairLengthSide}
       Hair Length on back: ${widget.hairLengthBack}
       Hairline: ${widget.hairLine}
       have soft lighting to emphasize the hair. The background should be a simple
       soft color to ensure the focus remains on the person's head.
       create four separate images
       should have: the left side angle
       should have: the right side angle
       should have: the front side angle
       should have: the back side angle
      ''';

  final request = GenerateImage(
      model: DallE3(),
      prompt,
      1,
      size: ImageSize.size1024,
      responseFormat: Format.url);  // GenerateImage
  try {
    final response = await openAI.generateImage(request);

    setState(() {
      isGenerating = false;
      urls = response?.data?.map((data) => data?.url ?? '').toList() ?? [];
    });
  } catch (e) {
    setState(() {
      isGenerating = false;
      urls = ['Error generating images: $e'];
    });
  }
}
```

Figure 7. Screenshot of code 3

The code is a function called generateImage() and it helps create pictures of a male's hairstyle using AI. It first creates a prompt variable, which is the set of instructions that will be telling the AI model what kind of hairstyle to generate. The prompt was carefully engineered through prompt engineering and trial and error. The prompt includes details like hair texture, hair length on all sides, and lighting. The program then creates a request using the dall e3 model to generate an image. The try block sends this request to the AI server and waits for a response [13]. If the response has no errors, it takes the image link and displays it on the generated image screen and saves it to a list called urls. If something goes wrong, the catch block catches it and shows an error message. This function runs when the user submits the inputs to the personalized hairstyle screen. The AI model's job is to read the prompt, create pictures, and send them back to the program.

4. EXPERIMENT

4.1. Experiment 1

A potential blind spot in the program is in the personalized image generation screen. This is driven by the AI model DallE3 and it is important that the prompt for the AI model works well because the prompt is the only information the AI gets to generate an accurate image. By minimizing blindspots through the prompt, results will meet expectations.

An experiment will be set up to change the prompt to see how changing the prompt affects the AI's images. The part of the prompt that will be changed will not be the hairline, hair texture, or hair length because these are essential information for the AI and should remain constant. However, the experiment will be set up so that the other words in the prompt will be changed to see which combination of words will allow for the best image displaying four angles of the hairstyle while still emphasizing the actual hair. Control data will come from the first set of images made with the original prompt. This will allow for comparison to the new images with a new prompt and old images with an old prompt. This way, it is clear to how AI reads and follows certain instructions.

Tested prompt	Expected Result	Real Result
Prompt 1: have soft lighting to emphasize the hair. The background should be a simple soft color to ensure the focus remains on the person's head. create four separate images should have: the left side angle should have: the right side angle should have: the front side angle should have: the back side angle	9	7
Prompt 2: Focus on hair four images: left right front back	4	2
Prompt 3: Simple background focusing on hair create four separate images: left side angle right side angle front side angle back side angle	6	5
Prompt 4: Use soft lighting to highlight the texture, volume, and flow of the hair. The background should be a smooth, neutral pastel shade (such as light beige, soft pink, or pale blue) to keep the focus entirely on the hair and head shape. Ensure that the lighting is diffused, avoiding harsh shadows while adding depth and dimension. The subject's hair should be well-defined with a natural and realistic appearance. create four separate images: should have: the left side angle should have: the right side angle should have: the front side angle should have: the back side angle	8	5
Prompt 5: Create four distinct images capturing a person's head from different perspectives: left profile, right profile, front view, and back view. Utilize gentle, diffused lighting to accentuate the hair's texture, movement, and volume without creating harsh shadows. The background should be a soft, muted pastel shade—such as light peach, pale lavender, or cool gray—to maintain focus on the subject's hair and head structure. Ensure the hair appears detailed and natural, with individual strands subtly visible to enhance realism and depth.	9	4
Prompt 6: Create four individual images showcasing a person's head from different angles: left side, right side, front, and back. Use soft, even lighting to highlight the hair's texture, shine, and natural flow while avoiding harsh contrasts. The background should be a smooth, neutral pastel tone—such as soft cream, pale blue, or muted pink—to keep attention on the subject's hair and head shape. Ensure the hair appears detailed and lifelike, with subtle variations in strands to enhance depth and realism.	9	5
Prompt 7: Generate four distinct images featuring a person's head from the following angles: left side, right side, front, and back. Utilize soft, diffused lighting to enhance the hair's texture, depth, and natural movement without harsh shadows. The background should be a simple, muted pastel color—such as light beige, pale lavender, or soft gray—to ensure the focus remains on the hair and head structure. Capture fine details in the hair, including subtle variations in strands, for a realistic and visually appealing effect.	9	6
Prompt 8: Produce four detailed images of a person's head, captured from four angles: left side, right side, front, and back. Employ soft, diffused lighting to emphasize the hair's texture, volume, and natural flow while avoiding harsh shadows. The background should be a smooth, understated pastel tone—such as light taupe, pale blue, or soft peach—to ensure the focus remains on the hair and head structure. Render the hair with lifelike detail, showcasing individual strands and subtle variations for a realistic and visually appealing effect.	9	4
Prompt 9: Create four distinct images of a person's head from different angles: left profile, right profile, front view, and back view. Use soft, natural lighting to gently illuminate the hair, emphasizing its texture, movement, and shine. The background should be a simple, soft pastel hue—like pale mint, soft lavender, or light gray—to keep the focus on the person's hair and facial structure. Ensure the hair appears realistic with delicate details, such as individual strands and natural flow, creating depth and dimension.	9	5
Prompt 10: Generate four separate images of a person's head, each from a different angle: left side, right side, front, and back. Use gentle, diffused lighting to bring out the hair's texture, highlighting its volume and natural movement. The background should be a light, neutral pastel color—such as soft peach, pale green, or light gray—to keep the focus on the person's head and hair. Ensure the hair looks realistic, with subtle details like individual strands and natural flow to add depth and dimension.	9	5

Figure 8. Table of experiment 1

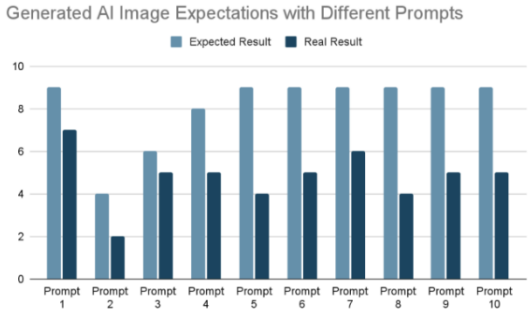

Figure 9. Figure of experiment 1

The mean value of the data set is 4 and the median is 4.5. The lowest value of the data set is a 2 and the highest value is a 5. A part of the data that can come off as surprising is how as the more specific the prompt gets, the expected result matching the real result doesn't become better complemented. The data that didn't meet expectations was the last prompt or the most specific one. The expectation for the most specific prompt was the most accurate image matching the expected image. It turned out this way because by getting too specific with the AI model, the AI model will have some blindspots and emphasize not as relevant parts more than the actual relevant ones. The biggest effect of this on the results is the difference between the expected picture and how well it matched the actual picture.

4.2. Experiment 2

A potential blindspot in the program is the Zip Codes that don't exist in the database and therefore creating a blindspot where users living in these zip codes aren't able to find barbers near them using the program. It is important because by knowing what percentage of zip codes don't work, a disclaimer can be given to the user to uphold satisfactory.

To test the percentage of missing zip codes there are in the program to the amount of zip codes that exist, the experiment will continuously test different zip codes found on a valid list of U.S. zip codes sourced from the USPS. By identifying which zip codes are missing, the percentage of unrecognized zip codes can be calculated through dividing the amount of unrecognized zip codes by the amount of zip codes tested. This experiment is structured to ensure accuracy by comparing the database inside the barber app to a well known and verified database. Control data will be sourced from the USPS zip code database.

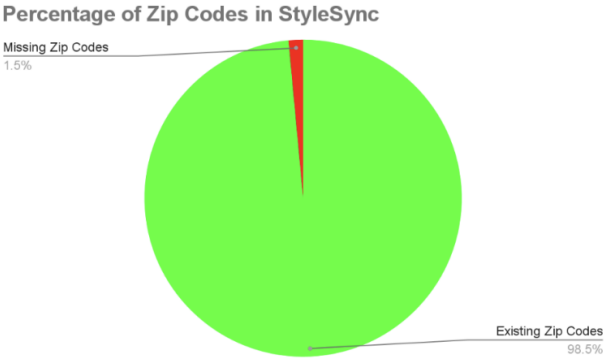

Figure 10. Figure of experiment 2

The pie chart shows that 98.5% of the zip codes in Booksy exist while 1.5% are missing. The amount of zip codes tested is 200 and out of 200, only 3 were missing. Thus, the lowest value is 3 while the highest value is 197. The mean and median both lean towards existing zip codes as it is the clear majority. The low percentage of missing zip codes suggests that the program's database contains a suitable amount of zip codes to satisfy the user. If a higher missing rate was expected, this result might be surprising. Possible reasons for a higher missing rate than expected includes non regular updates and use of an unverified database. The biggest factor affecting the results is the completeness of the original dataset. If it was sourced from an unreliable database, the results would converge to a higher percentage of missing zip codes.

5. RELATED WORK

The solution that StyleSage came up with to personalize hairstyle and recommend hairstyle is through a machine learning-powered hairstyle recommendation tool that uses facial recognition and face shape classification to most suit a user [4]. This solution is effective as it provides the user with a closely personalized hairstyle, but this raises privacy concerns as it could be too personalized. For example, some limitations of StyleSage is that users and developers would have to take into account the user's face being uploaded online. In contrast, StyleSync doesn't rely on facial recognition to personalize hairstyle and therefore avoids the privacy concerns that StyleSage carries with it.

The article "Web Scraping Techniques and Applications: A Literature Review" outlines how web scraping is used to collect real-time data from online platforms including directories [5]. While this solution is very effective for gathering barber profile data such as location, reviews, and services, it still has its limitations. For example, web scraping is something that can be hard to control as it can grab too much unnecessary information that confuses the user or makes the app hard to read. StyleSync improves on this by filtering and cutting the information that was gathered through web scraping ensuring only the expected information is received and displayed.
The article "Recommendation System for Hairstyle Based on Face Recognition Using AI and Machine Learning" explores how AI and machine learning models use face recognition to recommend hairstyles based on users facial features [6]. This method is effective in providing closely personal suggestions, but at the same time, faces some limitations. For example, the heavy reliance on facial recognition technology. Facial recognition technology, like any other technology, has its limitations and is not 100% accurate and these limitations carry over to this method. In contrast, StyleSync doesn't use facial recognition at all to gather data for the AI model, but it uses specific information imputed by the user. This removes the probability of errors which was present in the article's solution.

6. CONCLUSIONS

One limitation of the project is that the AI-generated hairstyle recommendations may not always be accurate or realistic, especially if the model doesn't fully account for details like hair density, texture variation, or natural growth patterns [14]. Additionally, relying on users to input personal hair information can lead to inaccuracies if they are unsure about their hair type or face shape. Privacy is another concern, as storing personal images, preferences, and location data requires strong data protection [15]. The app may also struggle in areas with fewer barbers, limiting its effectiveness in connecting users to local professionals. Another issue is that AI models can reflect bias if they are not trained on diverse datasets, potentially underrepresenting certain hair types or cultural styles. With more time, we would improve accuracy using facial and hair analysis powered by computer vision, expand our dataset for diversity, integrate privacy-first data handling, and create tools for barber engagement and real-time consultation.

In conclusion, this app offers a promising solution to modern hairstyling needs by combining personalization, AI, and accessibility. Addressing current limitations will enhance user experience, ensure inclusivity, and strengthen security. With continued development, the app can become a powerful tool in transforming how users choose and book their hairstyles.

REFERENCES

[1] Andrist, Linda C. "Media images, body dissatisfaction, and disordered eating in adolescent women." MCN: The American Journal of Maternal/Child Nursing 28.2 (2003): 119-123.

[2] Brand, Horst, and Ziaul Z. Ahmed. "Beauty and barber shops: the trend of labor productivity." Monthly Lab. Rev. 109 (1986): 21.

[3] Staten, Vince. Do bald men get half-price haircuts?: In search of Americas' great barbershops. Simon and Schuster, 2001.

[4] Chinnasamy, P., et al. "Hair Style Recommendations using CNN and AR." 2024 9th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2024.

[5] Khder, Moaiad Ahmad. "Web scraping or web crawling: State of art, techniques, approaches and application." International Journal of Advances in Soft Computing & Its Applications 13.3 (2021).

[6] Kamble, Yogesh M., and Raj B. Kulkarni. "Recommendation System for Hairstyle Based on Face Recognition Using AI and Machine Learning." International Journal of Software Innovation (IJSI) 12.1 (2024): 1-10.

[7] Mohanapriya, T., et al. "Impact of digital technologies in health care sector with reference to AI-Powered applications." AIP Conference Proceedings. Vol. 3161. No. 1. AIP Publishing, 2024.

[8] Shah, Chirag. "From prompt engineering to prompt science with human in the loop." arXiv preprint arXiv:2401.04122 (2024).

[9] Mesko, Norbert, and Tamas Bereczkei. "Hairstyle as an adaptive means of displaying phenotypic quality." Human Nature 15 (2004): 251-270.

[10] Feuerriegel, Stefan, et al. "Generative ai." Business & Information Systems Engineering 66.1 (2024): 111-126.

[11] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." Future Internet 15.6 (2023): 192.

[12] Fergus, Robert, Pietro Perona, and Andrew Zisserman. "A visual category filter for google images." Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8. Springer Berlin Heidelberg, 2004.

[13] Xu, Guangquan, et al. "Real-time diagnosis of configuration errors for software of AI server infrastructure." IEEE Transactions on Dependable and Secure Computing (2023).

[14] Birch, M. P., J. F. Messenger, and A. G. Messenger. "Hair density, hair diameter and the prevalence of female pattern hair loss." British Journal of Dermatology 144.2 (2001): 297-304.

[15] Bygrave, Lee A. "Privacy and data protection in an international perspective." Scandinavian studies in law 56.8 (2010): 165-200.