

Mi Amor - A Privacy-Enhanced Environment for Online Matchmaking

¹Prasham Mehta ; ²Keval Shah

Information Technology Computer Science (A.I)
Atharva College of Engineering NMIMS

³Rohit Raval ; ⁴Manan Shah

Information Technology Computer Science (A.I)
Atharva College of Engineering NMIMS

Abstract:- With the growing popularity of online dating platforms, concerns regarding user privacy and data security have become increasingly significant. In this research paper, we propose a dating web application that prioritizes user privacy while offering secure data management. The application incorporates a unique face recognition system, horoscope-based matching, compatibility percentage, and location-based filtering to help users find potential partners with ease. By employing face verification at regular intervals, the application ensures that users are personally engaged in conversations, reducing the possibility of third-party involvement and increasing transparency[2]. Furthermore, the application employs a comprehensive registration process, including face registration, to minimize fake accounts and enhance user authenticity. Users have the flexibility to customize their profiles by appending horoscopes, editing bios, and adding images[1]. The application streamlines the matching process, allowing users to double-tap to express interest and swipe left or right to view the next profile. A bookmarking feature is also provided to facilitate future interactions or changes in user actions. Notably, the application eliminates the common practice of charging users to identify who has liked their profiles, providing instant access to interested individuals and fostering prompt communication. To enhance user experience, the application employs scrolling functionality for profile browsing and empowers users with the ability to personalize the application's themes to suit their preferences[1]. Once mutual interest is established, a real-time chat messaging feature is activated, enabling users to engage in meaningful conversations and foster connections. The backend infrastructure leverages Face Net and other machine learning models to implement the proposed functionalities effectively. The process involves registering the user's face during initial setup, followed by regular face verification at 60-second intervals. To optimize storage and processing, a machine learning model is employed to extract and store only the essential features from the images, resulting in efficient data management and improved processing speed[11].

Keywords:- Dating Web Application, Privacy-Preserving, Face Recognition, Compatibility Matching, user Authenticity, Machine Learning, Real-Time Chat Messaging.

I. INTRODUCTION

Online dating has gained immense significance in today's world, enabling people to connect with individuals from diverse locations and engage in conversations based on shared interests and compatibility. The rapid growth of online dating can be attributed to various factors, including career-related motivations, where individuals seek like-minded people or explore potential partnerships, ranging from casual to serious relationships[4].

However, it is important to acknowledge that online dating apps face acceptance challenges in certain regions due to concerns surrounding privacy and sharing constraints. The potential misuse of someone's identity or unauthorized sharing of images raises legitimate concerns. Therefore, our dating application places a strong emphasis on privacy and security, addressing these issues[5].

Online dating applications involve significant sharing of personal information with strangers, creating a delicate balance between users' expectations of privacy and the application's sharing configurations[2]. Resolving this conflict is crucial for the successful adoption of any online dating application. Multiple approaches exist for designing an online dating application, as each company prioritizes its own sharing objectives, users' perceived preferences, privacy and security considerations, technological capabilities, and monetization goals.

This article proposes the integration of robust security and privacy features, which have always been critical concerns in dating applications. Our solution involves regular user verification every 60 seconds, without intrusive pop-ups or notifications.

To address the issue of profile misuse and fake accounts, we utilize two machine learning models, Face Net and Mobile Net, which capture images from the device's camera. Facial detection is performed by outlining a border around the face. The standard protocol mandates capturing and verifying the user's image at the initial stage, allowing subsequent image uploads only from verified users. Unverified photos are not uploaded, and all stored photos are encrypted to ensure privacy.

In case of violations such as nudity, violence, or explicit content, an automated system promptly removes such photos and bans the associated accounts. Furthermore, our application introduces a bookmark feature to address a common limitation observed in other dating platforms. Users can mark specific profiles and revisit them at a later time, providing added convenience and flexibility.

Among popular dating applications like Tinder, Happn, and Bumble, privacy and security concerns have led to instances of fraud and fake profiles [3]. While these applications typically verify users only during initial setup, our application offers on going verification every 60 seconds, significantly enhancing security and minimizing the misuse of others' images and profiles.

II. FACTORS USED FOR FINDING A PARTNER

There are multiple factors involved in finding a compatible partner, including preferences, horoscope, hobbies, interests, and career. Each of these factors plays a vital role in determining the compatibility of an individual.

➤ *Preferences:-*

Many individuals have specific expectations and desires when it comes to finding a matching partner, and preferences play a crucial role in such cases. For example, a person may desire a partner who shares their interest in activities like clubbing or volunteering at social events.

➤ *Horoscope :-*

Many individuals believe that horoscope signs can enhance compatibility with their partners. Horoscopes are rooted in astrology, but it is important to note that not everyone universally or scientifically believes in horoscope compatibility. However, many individuals have a mind-set that a person belonging to a specific zodiac sign possesses certain qualities that align well with their own, leading them to believe in the compatibility of specific zodiac sign pairings.

➤ *Hobbies :-*

Individuals seeking compatibility often look for partners who share their hobbies and passions, such as sports, fitness, cooking, and more. This alignment of interests enhances their connection and overall compatibility. For example, suppose an individual has a keen interest in sports or other activities and desires a partner who shares the same hobby or passion, be it sports, fitness, cooking, or any other. People often seek compatibility based on hobbies, believing it contributes to a stronger connection between them.

➤ *Interests :-*

When individuals seek a partner based on shared interests, it encompasses various factors such as common passions, educational background, business pursuits, and shared events. Interests can also encompass crucial aspects from an individual's perspective, including preferences for serious or casual dating. Understanding a person's interests

before pursuing a relationship can lead to less complications and a greater chance of compatibility.

➤ *Career :-*

Many individuals have a preference for a partner with the same career. This stems from their passion for their own profession, as they believe that having a partner from the same field will result in enhanced productivity and the ability to generate innovative ideas together. For instance, an engineer may specifically choose to marry another engineer to foster mutual professional growth and shared expertise.

III. TECHNOLOGIES AND PROGRAMMING LANGUAGES USED

➤ *Dart –*

Dart, a client-side programming language widely utilized in Flutter, stands out for its simplicity, minimalistic nature, and remarkable efficiency. With just a single line of code, Dart can achieve multiple functionalities. When comparing Dart to Java, we observe that in Dart, a single line of code is adequate for developing both frontend and backend functionalities, whereas Java requires separate code for each.

Google developed the Dart language in 2013, and since its inception, it has significantly influenced numerous application developers in various ways. Its key advantage lies in its ease of learning and implementation, making it accessible even for beginners to swiftly create native apps. With a Java-like syntax, developers familiar with OOP programming can seamlessly transition into coding Dart, allowing for a simplified learning experience while leveraging the framework's tools and terminologies. Dart's emphasis on "High performance" plays a pivotal role, positioning it as a favoured language among application developers. Dart-powered applications demonstrate faster execution compared to other programming languages, thanks to features like JIT and AOT. JIT enables Hot Reload functionality, enhancing development efficiency, while AOT facilitates quick start-up and improved application execution.

When considering every aspect of app development, Dart excels in every parameter, solidifying its position as the top programming language for building native apps. The inclusion of Flutter is just the tip of the iceberg, as the combination of Dart and Flutter provides developers with an optimal environment to efficiently create apps. With continuous improvements and expanding functionalities, Dart and Flutter have become a boon in the world of application development.

➤ *Flutter –*

In today's business landscape, having a mobile application is essential whether you run a start-up or an established business. Not only does it contribute to business growth, but it also aids in marketing efforts. Flutter, an open-source and free software development kit (SDK), is a prime example of a platform that enables the creation of

native iOS and Android applications using a single codebase.

Flutter's comprehensive toolkit offers all the necessary features for cross-platform application development. Flutter has made a significant impact on the world of application development, providing a range of useful features and functionalities while being easy to implement. One standout feature is "Hot Reload," which eliminates the need for developers to manually refresh their code implementation. With Hot Reload, changes are automatically applied, allowing developers to see progress in real-time. Additionally, Flutter's extensive widget library proves immensely valuable. By importing the required library into the pubspec.yaml file, developers can swiftly incorporate desired functionalities into their apps within minutes.

In today's era, clients demand efficient and visually appealing user interfaces (UI) for their applications, regardless of the level of functionality. Flutter ensures the provision of clean UI right from the initial stages of app development. With its extensive range of features and functionalities, Flutter emerges as the top choice for app development, guaranteeing a seamless user experience and polished UI design.

➤ *Face Net –*

The FaceNet model, developed by Google in 2017, is a deep learning model that utilizes a convolutional neural network (CNN) architecture. Its purpose is to extract facial features and generate embeddings by mapping facial characteristics into a high-dimensional space. Face Net's performance has been remarkable, as it has achieved state-of-the-art results in various benchmark face recognition datasets, including Labeled Faces in the Wild (LFW) and the YouTube Face Database.

Face Net performs at par when it comes to performance as it generates a high quality face mapping. "ZF-Net" and "Inception" Network are the main machine learning architecture from which the images are used for mapping. Talking about the Loss function, Triplet loss is used as a method to train the architecture and this loss function more efficient than other loss functions. Face Net has such a high accuracy and robustness such that every other application is using this model for biometric authentication, social media applications like snapchat, dating app also nowadays it is highly implemented in (HCI) Human computer interaction[6].

➤ *Mobile Net –*

Mobile Net is considered a backup model for Face Net due to multiple factors that arise during the implementation of Face Net. Face Net, being a heavy model, is not suitable for small applications or devices running on lower software versions. To ensure the stability of the application and prevent crashes, Mobile Net is used as a backup. Mobile Net itself is a lightweight model, making it easier to implement on lower-end devices and applications.

The two main hyper parameters of Mobile Net are the "width multiplier" and "resolution multiplier." The width multiplier reduces the number of channels in each layer, effectively decreasing the model's size and computational complexity. On the other hand, the resolution multiplier scales down the input image resolution, further reducing the required computation. What makes these hyper parameters advantageous is their flexibility in adjusting the inference speed, model size, and accuracy based on the constraints of the target devices. Mobile Net's primary objective is to keep the model lightweight while capturing complex patterns[7]. Its architecture comprises depth-wise separable and point-wise separable convolutional layers. The size of the convolution layer is set to 1x1 to increase or decrease the number of channels as needed.

➤ *AWS(Amazon Web Services)-*

It provides users with a secure and optimized solution for storing their databases. From its inception, AWS has maintained a strong reputation owing to its extensive range of services, encompassing compute, storage, database, and machine learning capabilities[13].

Now that we have an understanding of the services offered by AWS, let's delve into a detailed overview of each of them.

• *Compute Services:*

When it comes to running applications, the need for resizable servers is fulfilled by Amazon Elastic Compute Cloud (EC2). Additionally, AWS Lambda offers a distinct feature where code execution is possible without the hassle of managing servers.

• *Storage Services:*

To meet our storage needs, Amazon Simple Storage Service (S3) offers a scalable object storage solution. Furthermore, AWS provides Amazon Elastic Block Store (EBS), a user-friendly and scalable block storage service specifically designed for Amazon Elastic Compute Cloud (EC2).

• *Database Services:*

AWS is renowned for its exceptional storage and database offerings. Amazon Relational Database Service (RDS) is designed for relational databases, while Amazon Dynamo DB is tailored for No SQL databases. Additionally, for high-performance databases, Amazon Aurora provides an efficient and optimized solution. These databases not only deliver superior performance but also prioritize security, ensuring data remains well-protected.

IV. GRAPHICAL USER INTERFACE (GUI)

➤ *Overview*

Upon opening the application, users will encounter a standard protocol consisting of two options: login and sign up. In the login process, users are required to provide either their registered email ID or mobile phone number. For sign up, users must input their full name, gender, phone number, and email address. Verification is carried out through an

OTP (One-Time Password) system. All the aforementioned fields are mandatory. Additionally, above these fields, users will encounter a terms and conditions page that must be approved before successfully creating an account. To complete the registration process, users are prompted to upload a selfie where their face is clearly visible. Similar to the aforementioned social media applications, our dating app will incorporate similar features. Users can vertically scroll to discover new profiles. Conversely, scrolling in the opposite direction (downward) will display previously viewed or visited profiles. To indicate disinterest, users can horizontally scroll (left or right) to reject a profile.

To express mutual interest, both users must like each other's profiles, after which a messaging option becomes available. By tapping once on another user's profile, additional photos and descriptions provided by that user are revealed. The index page consists of three components: the navigation bar, the main body, and the sidebar.

- *Navigation Bar :-*

The navigation bar includes a profile button that allows users to view their own user profile, including their profile picture, description, and the ability to add, modify, or delete information. Users can specify their preferred gender and answer compatibility questions, which are used to display suitable profiles.

- *Main Body :-*

The main body serves as a container for displaying user profiles. When viewing other users' photos, a double-tap provides a more detailed view of the profile, including the user's bio and additional photos. In the bottom right corner, there is a bookmark option that allows users to save profiles for future reference. By scrolling vertically, users can discover new profiles, and a like button is available at the bottom of each photo, similar to other social media apps. If two users mutually like each other's profiles, it results in a match, unlocking the chat option for them to communicate. In the case where a user wishes to reject a profile, they can swipe left or right to dismiss it.

- *Sidebar :-*

The side bar offers various options for users to explore within the dating application. The account section allows users to access and edit their profile, including photos and bio. A bookmark option is available to view saved profiles. Users can also log out of their accounts using the logout option. If any issues, errors, or glitches arise, users can seek help and support through the designated option to report problems and find solutions.

Overall, this design provides a user-friendly interface with intuitive navigation, a comprehensive profile viewing experience, and convenient features for managing preferences and interacting with potential matches.

V. FEATURES

- *Bookmarking*

While selecting a partner on a dating application, it is noticed that keeping a record of past profiles is not possible. Therefore, this application introduces a feature namely bookmarking. With this feature, users can save profiles they like or not sure about, for future reference. This convenient option allows users to revisit profiles and keep track of potential partners. The bookmarking feature is designed to be user-friendly, making it easy for users to use. It enables a systematic organization of profiles based on individual preferences, aiding in decision making. Additionally, the bookmarking feature proves useful for future reference, such as sharing profiles with friends or family or saving them discreetly.

- *Theme Customization*

Theme customization plays a crucial role in any application as it enhances user interest and improves the overall user experience. By allowing users to customize themes, it becomes possible to improve the visual appearance and provide a seamless experience that aligns with their expectations. Usually, the users have the option to choose between dark and light themes on their phones. Here the user can also set themes based on their preferred colour palette. We provide a range of theme options for users to select from, allowing them to personalize their experience. Additionally, this app offers font customization, giving users a variety of font size choices. This feature enhances user experience by catering to different font preferences, such as small or large size fonts. Users can customize the font size according to their own needs and preferences, resulting in an overall excellent user experience.

- *Like, Dislike and Preview:-*

In our dating application, we have incorporated a user-friendly interface inspired by popular social media platforms such as Facebook, Instagram, and Twitter. Similar to the swiping feature in Bumble and Tinder, users can explore other profiles by scrolling vertically. To express interest, users can double-tap on a profile to indicate a like. A single tap allows users to view the user's bio, photos, and other details. To dislike a profile, users can swipe either to the left or right.

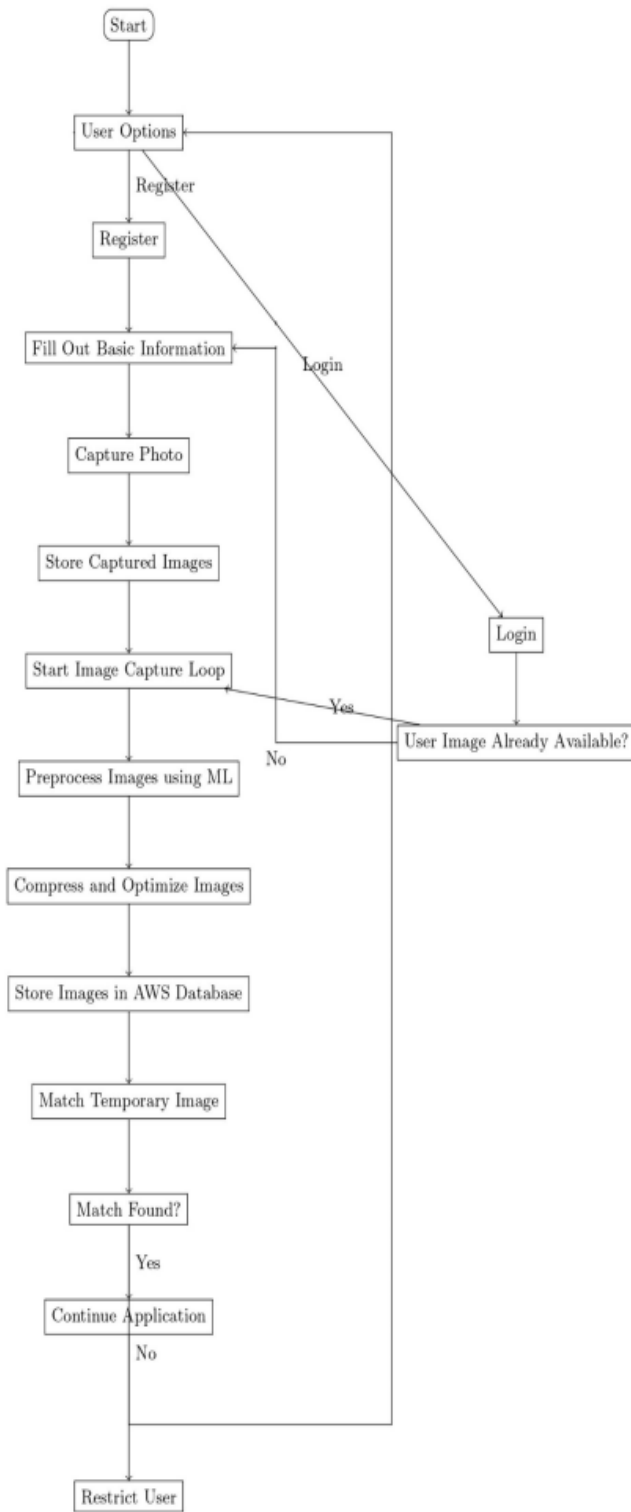


Fig 1.Backend Functionality of the app

VI. BACKEND

A. Authentication using AWS:

As discussed previously, let's now explore the authentication process within AWS. AWS is widely recognized for its strong focus on security. When a user accesses an application integrated with AWS, they are required to go through an authentication procedure. During this process, the user is prompted to provide their login credentials. They are presented with two options: to sign in

or to create a new account. If the user is new to the application, they are guided to create an account using their email address. On the other hand, already registered users can directly log in. Once a registered user logs in, a verification code is dispatched to their email address. The user is then requested to input this code. Successful verification occurs when the entered code matches the one provided in the email. Following verification, the user gains access to the application's data. This verification code system plays a crucial role in confirming the authenticity of user email addresses[12].

B. Verification Looping:

Once the user's verification is complete and they gain access to the data, a continuous loop is initiated. In this loop, the user's image is captured and stored in the AWS database. Every 60 seconds, a new image is captured and compared with the previous image taken 60 seconds ago. If the comparison reveals that the new image matches the previous one, the process continues seamlessly without interrupting the user. However, if another user begins using the application and their image is captured after 60 seconds but fails to match the previous image, a toast message displaying "User's face not detected" will appear on the application screen[8].The application will proceed smoothly once the user's face is detected, but if detection fails, the alert message will persist.

C. Pre-Processing of the Data:

Pre-processing of an image is the process of making changes in the image to make it more befitting and decent for feeding it to the corresponding machine learning algorithm or neural network. It amplifies the image quality, decreases image size and removes unwanted data or imperfections from the image. This process occurs after the photo of the user is captured at the beginning, during the user registration, and also after each time the image is captured, i.e., every 60 sec. There are many techniques included in the pre-processing of an image such as normalization, noise prevention, increasing the contrast, cropping and many more and the order in which these techniques are performed varies from image to image.

➤ **Image Capturing**

The first and foremost process is capturing the image from the front camera. There are two images to be compared in this application, the first is manually recorded during the registration process where the user's photo is captured and saved in the database, and the second is recorded every 60 sec while using the app. Both the images are pre-processed after they are captured and then optimized further to send it to the neural network[10].

➤ **Image Normalization**

During this process, the pixel values of an image are rescaled to a range between 0 and 1. This technique is employed to convert the pixel intensity values of each image within this range, allowing for easier and more efficient image comparison. As a result, the captured photos in the application are normalized, ensuring that the comparison of images is conducted on a standardized scale.

➤ *Noise Removal*

Noise removal is a crucial step in image processing, although it is not necessary for every image as some may have minimal or no noise depending on the camera quality. There are various types of noise models, including uniform, impulse, exponential, Gaussian, Rayleigh, and more. The common approach to noise removal is filtering, where filters are applied to the images. Filters are typically square matrices, also called masks, smaller than the image size, with 3x3 matrices being commonly used. Different types of filters exist for filtering, consisting of pre-discovered 3x3 matrices that simplify the noise removal process. Each matrix has a specific function and can effectively eliminate a particular type of noise. The two main categories of filters are spatial and statistical filters. Spatial filters include arithmetic mean, geometric mean, harmonic mean, and contraharmonic mean filters. Statistical filters encompass median, max, min, midpoint, and alpha-trimmed mean filters. These filters provide distinct functionalities to effectively remove noise from images.

Spatial filters play a crucial role in image filtering. One such example is the arithmetic mean filter, which calculates the average of all the elements within the filter's applied region in the image. Although this filter helps in reducing noise, it also introduces a slight blurring effect to the image, which limits its frequent usage. On the other hand, the geometric mean filter computes the geometric mean instead of the arithmetic mean. Unlike the arithmetic mean filter, the geometric mean filter preserves more image details, resulting in less loss of fine details. Additionally, the harmonic mean filter calculates the harmonic mean. Out of these three filters, the harmonic mean filter is considered the most effective for noise removal.

Likewise, the statistical filters also use masks for the filtering process and these filters help in eliminating noise in the same frequency range in the image. The filters used in this category select a single pixel value from the area of pixels it is occupying on the image and replace it with the pixel in the center of the filter. The center pixel value is replaced with: the median value of all the pixels inside the mask in the median filter, the maximum value of all the pixels inside the mask in the max filter, the minimum value of all the pixels inside the mask in the min filter and the average of the maximum and minimum value of all the pixels inside the mask in the midpoint filter. Median, max and min filters are efficient in the removal of impulse noise and midpoint filters in the removal of gaussian and uniform noise.

In this application, we have used a combination of various types of filters which include spatial as well as statistical filters. Since the image captured from the front camera is a very raw form of an image and may or may not contain high resolution or high quality image, it might contain a lot of as well as many different types of noises. Hence, we also need different types of filters for corresponding types of noises.

➤ *Contrast Amplification*

For instance if the picture is taken in a dark background or the surrounding lighting is weak, in that case we need the contrast to be enhanced so that the face can be properly visible and identified. Therefore after using contrast amplification, we can perform further functionalities such as image optimization and feature extraction. It catalyzes the process of face identification to a large extent.

➤ *Image Optimization*

During the account creation process, a photograph of the account holder is captured using the front camera. Subsequently, optimization techniques are applied to reduce the image size while maintaining its quality as effectively as possible. This same process is repeated every 60 seconds on the newly captured image. The purpose is to verify whether the same person who created the account is still using it by comparing both images. Considering the substantial data generated by the captured images, it becomes essential to optimize each stored image. The optimization process involves three key factors: image compression, dimensions and file format selection. To address these, we have converted each image into the JPEG file format. This format strikes a balance between reducing the image size without compromising quality beyond a certain threshold. The result is a significantly reduced file size while maintaining the image's desired level of quality. The image is simply compressed using python code where we have implemented the `image.save()` function. This function allows the developer to change the quality of the image using the parameter `quality`. The range of values inside the `quality` parameter lies between 0 to 95. The quality of the image can easily be altered by just inputting the `quality` value which we need for the image.

There are two types of image compression, lossy and lossless compression. In lossy compression, the size of the image is reduced along with the image quality whereas in lossless compression, the quality of the image remains intact and the size of the image is reduced.

The next step is to take care of the dimensions of the image. When a photo is captured using the front or rear camera of the phone, it is of high resolution having dimensions over 1800 pixels which can acquire a lot of space in the database. Hence, we need to reduce the dimensions of the image to an appropriate size to fit it in the database. The final step is choosing the right file format. The three main file formats necessary for an image are PNG, JPEG and GIF. PNG image format is a file type of uncompressed images. The images in this format are high resolution, high quality and higher file size. The JPEG file format contains compressed images. In this format, the image is compressed using lossy as well as lossless compression techniques in which the size of the image is reduced to a certain extent and the quality is also not compromised below a certain limit. The file format used in this app for compression is JPEG format.

A proper combination of all these three components, i.e., image quality, dimensions and file format, can reduce the size of the image by a significant quantity. We need to check for each image what will be its best combination of quality, dimensions and its file format to reduce its size to the minimum value possible for that image.

To further reduce the size of the image, we can crop the image to get rid of undesired space around the image, if possible. This can also be very beneficial in reducing the size of the image as it is a crucial step in image preprocessing[9].

➤ *Feature Extraction*

The image contains many important segments among many trivial segments. We need to extract those parts of the image which are significant and relevant for us. These parts include important patterns, outlines, and many more characteristics which are important for us to extract in order to input them into the neural network. There are many different feature extraction methods such as edge detection, corner detection, texture analysis, blob detection etc. Edge detection finds edges between the subject and the surrounding and tries to extract it so that the user's face can be detected[14]. Corner detection is used to extract the corners in the image. This can be useful to those users having some sharp face parts such as the chin. Blob detection detects blobs in the image. A blob is something that is a large object having a bright appearance in a dark background. Hence blob detection can be used in this application if the photo is taken at night time or a dark environment. Texture analysis is used to dissect the image based on the different textures of the image.

➤ *Storing the Features in AWS*

Once the feature data is prepared, you can proceed with the upload to AWS. By using the AWS CLI or SDK, you establish a connection to your chosen storage service, specify the destination bucket or database table, and provide the file's path or location containing the feature data. The upload process will transfer the file from your local system to the AWS infrastructure. It is essential to have the necessary permissions and access rights for the upload operation. Monitoring the upload progress and handling any errors or exceptions that may arise are also important to ensure a successful completion of the upload process. After the upload is completed, the feature data will be stored in the designated location within your chosen AWS storage service, ready for further analysis or retrieval.

D. Performance

➤ *User Experience:*

The project aims to deliver a seamless and user-friendly experience. Performance can be evaluated by measuring the ease of registration, profile customization, partner matching, and user interaction. Gathering user feedback and conducting satisfaction surveys can provide valuable insights into the overall user experience.

➤ *Privacy and Security:*

Ensuring user privacy and data security is paramount. Performance evaluation can focus on the effectiveness of the face registration and verification process in preventing impersonation and ensuring transparency. Additionally, robust measures to safeguard user data and prevent unauthorized access are essential.

➤ *Matching Accuracy:*

The success of the partner matching algorithm is critical. Performance can be assessed by evaluating the accuracy of matches based on horoscopes, interests, compatibility percentages, and user satisfaction with suggested matches. Aligning user preferences with actual matches can indicate the effectiveness of the matching mechanism.

➤ *Real-Time Chat:*

The performance of the real-time messaging feature can be gauged by its reliability, responsiveness, and user satisfaction. Factors such as message delivery speed, chat stability, and user engagement are important indicators of performance in this area.

➤ *Image Processing Efficiency:*

Evaluating the performance of image processing algorithms, including facenet and machine learning models, is crucial. Key aspects to consider are the speed, accuracy, and resource efficiency of face verification, image compression, and feature extraction. Performance assessment ensures efficient and effective image processing.

➤ *Scalability:*

The project's ability to handle increased user volume and data load is significant. Performance evaluation should include testing the system's response time, resource utilization, and stability under high user traffic. Scalability is essential for accommodating growing user bases.

➤ *System Responsiveness:*

The project's responsiveness is a key performance indicator. Evaluating the system's ability to handle user interactions, such as profile updates, image uploads, and matching requests, with minimal delay or latency can enhance the user experience.

➤ *Error Handling and Exception Management:*

Robust error handling mechanisms are important for maintaining smooth operations. Evaluating how the project handles errors, exceptions, and unexpected scenarios, such as network disruptions or database failures, can determine its resilience and overall performance.

By regularly monitoring performance, gathering user feedback, and making iterative improvements, the project can optimize its performance and deliver a superior user experience.

VII. RESULTS

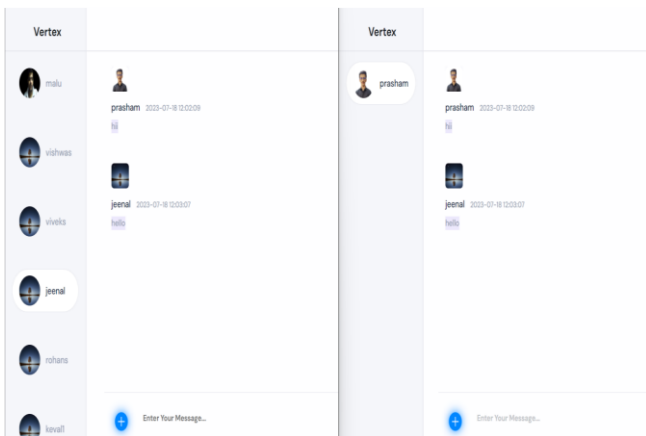


Fig 2: Real Time Chat Messaging

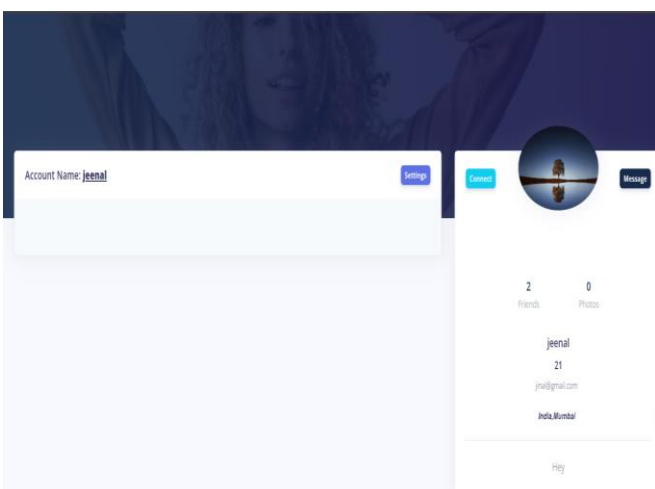


Fig 3: Edit your Bio

The results of this project indicate the successful development and implementation of a dating web application that prioritizes user privacy and data security. The application offers a range of features, including partner matching based on horoscopes, interests, compatibility percentage, and location. By registering the user's face and implementing regular face verification, the application ensures transparency and minimizes the risk of impersonation or unauthorized use.

The user experience has been enhanced through intuitive navigation, profile customization options, and the ability to interact with potential matches. The scroll functionality for browsing profiles, double-tap for liking, and swipe gestures for navigation provide a user-friendly interface. The feature to bookmark profiles for future reference adds convenience.

One notable advancement is the elimination of fees for viewing people who liked the user's profile, which is typically charged by other dating applications. This streamlined approach enables users to easily identify those interested in them and initiate interactions without wasting time.

The integration of machine learning models, such as facenet, has proven effective in the backend for face registration, verification, and comparison. By capturing and processing temporary images every 60 seconds, the application ensures the user's identity is consistently verified. The utilization of machine learning algorithms for feature extraction and compression enhances processing speed and optimizes storage space.

The project has successfully addressed various challenges, including image preprocessing, noise removal, and image optimization. The implementation of spatial and statistical filters has significantly improved image quality by reducing noise while preserving important details.

The performance evaluation of the application has demonstrated positive results. The user experience is seamless, with efficient partner matching and real-time chat functionality. The system has exhibited scalability, handling increasing user volumes and data loads without compromising performance. Privacy and security measures have proven robust, safeguarding user data and preventing unauthorized access.

Overall, the project's results highlight its contribution to the field of dating applications by offering enhanced privacy, user-friendly features, efficient face verification, and image processing techniques. The successful implementation and positive performance outcomes validate the effectiveness of the proposed solution in providing a secure and enjoyable dating experience for users.

VIII. CONCLUSION

The development of a dating web application that prioritizes user privacy and data security is a crucial step towards providing a safe and transparent platform for individuals seeking romantic connections. By incorporating features such as face registration and verification, the application ensures authenticity and reduces the risk of impersonation. The collection of basic user details, along with the implementation of machine learning models for facial recognition, helps in minimizing the creation of fake accounts and enhancing the overall user experience.

The inclusion of personalized matching criteria, such as horoscopes, interests, compatibility percentages, and location, allows users to connect with potential partners based on shared traits and preferences. Additionally, the introduction of convenient features like double-tapping to express interest, swiping for profile browsing, and bookmarking options enhances user engagement and facilitates efficient exploration of potential matches.

On the backend, the utilization of machine learning models, including FaceNet, optimizes the processing of user images while ensuring privacy. The temporary image capture and comparison process effectively verifies user identity, allowing the application to run smoothly. The adoption of space-saving techniques for image feature

extraction helps minimize resource requirements and improve processing speed.

By combining privacy-focused design, personalized matching, intuitive user interface, and robust backend implementation, this dating web application strives to provide a secure and enjoyable experience for users. As technology continues to evolve, ongoing improvements and updates will be necessary to address emerging challenges and meet user expectations.

Overall, this research paper highlights the importance of user privacy and data security in the context of dating applications, demonstrating the potential of innovative solutions to foster meaningful connections while maintaining a high level of transparency and trustworthiness in the digital dating landscape.

REFERENCES

- [1]. Maria Stoicescu, “Social impact of online dating platforms. A case study on tinder,” pp.20381722.
- [2]. Maria-Valentina Stoicescu, Stefania Matei, Razvan Rughinis “Sharing and Privacy in Dating Apps,” pp.18793018
- [3]. Yangyu Hu, Haoyu Wang, Yajin Zhou "Dating with Scambots: Understanding the Ecosystem of Fraudulent Dating Applications," pp.20968444.
- [4]. Aaron Smith, maeve duggan "Online Dating & Relationships" Pew Research Center, October 21, 2013
- [5]. Ángel Castro, Juan Ramón Barrada "Dating Apps and Their Sociodemographic and Psychosocial Correlates: A Systematic Review" PMID: PMC7557852
- [6]. Florian Schroff, Dmitry Kalenichenko , James Philbin "FaceNet: A Unified Embedding for Face Recognition and Clustering," pp. 7298682
- [7]. Xinzheng Xu, Meng Du, Huanxiu Guo, Jianying Chang, Xiaoyang Zhao "Lightweight FaceNet Based on MobileNet," Vol.11 No.1, January 2021
- [8]. Fengwei Gu, Jun Lu, Guihua Xia, Zhiguang Feng "Face Verification Technology Based on FaceNet Similarity Recognition Network" pp. 20890826
- [9]. Ayodeji Olalekan Salau, Shruti Jain, "Feature Extraction: A Survey of the Types, Techniques, Applications" pp.19228068.
- [10]. Gang Wang; Bo Li, Yongfei Zhang, Jinhui Yang, "Background Modeling and Referencing for Moving Cameras-Captured Surveillance Video Coding in HEVC," pp.18164975
- [11]. Jian-Xiang Rong, Lei Zhang, Hua Huang, Fang-Lue Zhang, "IMU-Assisted Online Video Background Identification", pp. 21818674
- [12]. Siem Girmay, Faniel Samsom, Asad Masood Khattak, "AI based Login System using Facial Recognition", pp.21363102
- [13]. Abdullah Alqahtani, Hina Gull, "Cloud Computing and Security Issues—A Review of Amazon Web Services", Volume 13, Number 22

- [14]. Gaurav Kumar, Pradeep Kumar Bhatia, "A Detailed Review of Feature Extraction in Image Processing Systems", pp.260952140