

“Akashic: The Smart Resource Management System”

Report submitted in partial fulfilment of the requirement for the

Degree of

B.Tech.

in

*Computer Science & Engineering (Artificial Intelligence and
Machine Learning)*

By

Ayush Agnihotri (2101641530049)

Ashutosh Pandey (2101641530046)

Abhraneel Singh (2101641530009)

Anish Singh (2101641530027)

Under the guidance of

Yashi Rastogi

Assistant Professor

Project Id: 22_AIML_2A_20



Pranveer Singh Institute of Technology, Kanpur
Dr A P J A K Technical University
Lucknow

DECLARATION

This is to certify that Report entitled “**Akashic: The Smart Resource Management System**” which is submitted by me in partial fulfilment of the requirement for the award of degree B.Tech. in Computer Science and Engineering to Pranveer Singh Institute of Technology, Kanpur Dr. A P J A K Technical University, Lucknow comprises only our own work and due acknowledgement has been made in the text to all other material used.

Date: 28/02/2023

Ayush Agnihotri (2101641530049)

Ashutosh Pandey (2101641530046)

Abhraneel Singh (2101641530009)

Anish Singh (2101641530027)

Certificate

This is to certify that the report entitled “**Akashic: The Smart Resource Management System**” which is submitted by **Ayush Agnihotri, Ashutosh Pandey, Anish Singh and Abhraneel Singh** in partial fulfilment of the requirement for the award of degree B.Tech. in Computer Science & Engineering to Pranveer Singh Institute of Technology, Kanpur affiliated to Dr. A P J A K Technical University, Lucknow is a record of the candidate own work carried out by him under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Signature:

Dr. Vishal Nagar
Dean CSE Department,
PSIT, Kanpur

Signature:

Miss Yashi Rastogi
Assistant Professor
CSE Department,
PSIT, Kanpur

ACKNOWLEDGEMENT

*It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. Second Year. We owe a special debt of gratitude to our project supervisor **Yashi Rastogi Assistant Professor**, Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur for his constant support and guidance throughout the course of our work. His sincere, thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavours have seen light of the day.*

We also take the opportunity to acknowledge the contribution of Professor Dr. Vishal Nagar Dean Computer Science & Engineering (Artificial Intelligence and machine learning) Department, Pranveer Singh Institute of Technology, Kanpur for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature

Name: Ayush Agnihotri

Roll No.: 2101641530049

Signature

Name: Ashutosh Pandey

Roll No.: 2101641530046

Signature

Name: Abhraneel Singh

Roll No.: 2101641530009

Signature

Name: Anish Singh

Roll No.: 2101641530027

ABSTRACT

This abstract describes a full-stack web-based Enterprise Resource Planning (ERP) system designed to integrate and automate business operations. The system is built with modern web technologies and comprises a suite of modules to manage various aspects of business operations, including inventory, sales, purchase, accounting, and human resources. The system is designed to provide a centralized platform to store and manage all the data related to business operations, and it enables real-time access to critical business information.

The ERP system is designed to provide a seamless experience for users, with an intuitive and user-friendly interface. The system is built using a responsive design, ensuring that it works seamlessly across different devices and platforms. The front-end of the system is built using modern JavaScript frameworks, while the back-end is built using a combination of PHP and Node.js. The system is also designed to be scalable, ensuring that it can handle large volumes of data and traffic.

Overall, the full-stack web-based ERP system is designed to provide businesses with a comprehensive platform to manage all their operations. It provides real-time access to critical business information, reduces the need for manual data entry, and improves the accuracy of data. The system is scalable and can be customized to meet the unique needs of businesses of different sizes and industries.

This report also details the development and evaluation of a machine learning face recognition module. The module utilizes a convolutional neural network (CNN) to extract facial features from images and map them to a high-dimensional embedding space. This embedding space is then used to compare and match faces for identification and verification purposes.

The CNN model was trained on a large dataset of facial images, including variations in pose, lighting, and facial expressions. The training process involved data augmentation techniques to increase the diversity and quantity of the training data. The resulting model achieved high accuracy and robustness in face recognition tasks, even under challenging conditions.

The evaluation of the module involved testing it on a separate dataset of faces, including individuals not seen during the training process. The results showed high accuracy in identifying and verifying faces, with minimal false positives and false negatives. The module was also evaluated for its ability to generalize to new datasets and demonstrated good performance.

In addition, the report discusses potential applications of the face recognition module, including security and

surveillance systems, access control, and personalized marketing. The ethical implications of using such technologies are also considered, including concerns related to privacy and potential biases in the training data.

Overall, the machine learning face recognition module presented in this report offers a powerful and reliable tool for identifying and verifying individuals in a variety of settings. Its performance and potential applications make it a promising technology for the future, while ethical considerations should be carefully considered and addressed.

TABLE OF CONTENTS

CHAPTER NO. TITLE	PAGE NO.
1. Introduction	9
2. Design Methodology.....	11
3. Implementation.....	14
4. Testing/ Result and Analysis	30
5. Conclusion and Future Enhancements	32
6. LIST OF FIGURES.....	8

LIST OF FIGURES

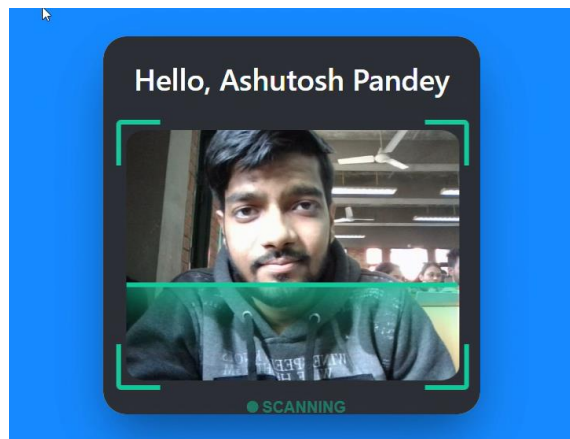
CHAPTER NO.	TITLE	PAGE NO.
1	Fig: - 1.1 Computer scanning the face of the user	9
2	Fig: - 2.1 Project Flow Diagram	11
2	Fig: - 2.2 Dataset Composition	12
2	Fig: - 2.3 Dataset training scale graph	12
2	Fig: - 2.4 Machine learning classifier architecture	13
3	Fig: - 3.1 Gantt chart of Project	14
3	Fig-3.2 Here's some example code to detect faces in an image using OpenCV	16
3	Fig-3.3 Here's some example code to pre-process a face image	18
3	Fig – 3.4 Here's some example code to extract and encode features from a pre-processed face image using a pre-trained CNN	21
3	Fig-3.5 Here's some example code to compare the user's face vector with their existing image in the database using the cosine similarity	22
3	Fig – 3.7 Akashic Landing Page layout	25
3	Fig – 3.7 Akashic Announcements Page layout	26
3	Fig: - 3.8 Computer scanning the face of the user	27
3	Fig – 3.9 Student Information	29

CHAPTER 1

Introduction (Scope of Work and its Importance)

1.1 As we see today, in the era of the Fast-moving world, everyone hesitates from typing credentials with long and hectic passwords which is sometimes tough to remember and there are chances of breaching of id and passwords. So, everyone wants a new, advanced and secure way of login that can make their login work faster and easier.

1.2 A new and advanced facial recognition login system. That will determine the face of the account owner and will secure login to the user panel by simply asking for a security pin for more security purposes. In case if the server is unable to determine the face due to some issues, then the account owner will be having a secondary method to recover his/her passwords.



(**Fig: - 1.1** Computer scanning the face of the user)

1.3 As we all know that; it won't be wrong to say that our life on digital media these days rely on passwords. Hence, while the purpose of passwords is to protect your account, we have reached that point in time where in return we are supposed to give the right protection to our passwords as well. But we are failing at that! So, our project Akashic will be working on its solution. According to the research conducted by HYPR in the last two and a half years all around the United States, India and Canada, the statistics show that internet users have been absolutely careless in managing their passwords; both in personal life and at work. Diving deep into the study, 78% of the 500 respondents accepted that they had to reset the passwords for one personal account in the last 90 days of their calendars. 57%, on the other hand, claimed that they had to go for password reset of their work account.

Another worrying statistic revealed that 65% of the users rely on digital and physical lists when it comes to

managing their password and despite the heavy advertisements, only 30% had password managers installed on their devices that also can be harmful for their Data Privacy. When users were obliged to reset the password, 49% of the respondents said that they used the same phrase with little changes here and there so that the password should remain easy to remember. Going even worse, 35% of the users keep their passwords in unprotected files such as Excel spreadsheets or text documents stored onto their computers. So, our Smart Login System will be a smart solution for this smart world. So, in this smart era of the Fast-Moving world, everyone needs a smart login system to prevent their hesitations of typing credentials, long and hectic passwords which are sometimes tough to remember and there are chances of breaching of ID and Passwords.

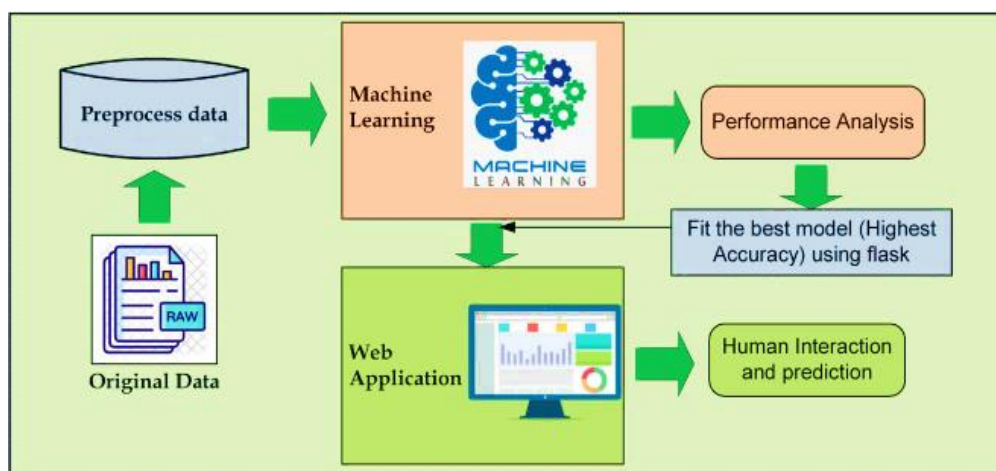
We are proposing a solution in the form of a new, advanced and secure facial recognition login system. That will determine the face of the account owner and will provide a secure login to the user panel by simply asking for a security pin for more security purposes. In case if the server is unable to determine the face due to some issues, then the account owner will be having a secondary method to recover his/her account

CHAPTER 2

Design Methodology

2.1 Dataset collection

2.1.1 It includes data collection and understanding the data to study the hidden patterns and trends which helps to predict and evaluate the results. Dataset carries more than 1500 rows i.e., total number of data and 10 columns i.e., total number of features. Features include Facial recognition, Gender determination, age recognition etc.



(Fig: - 2.1 Project Flow Diagram)

2.2 Data Pre-Processing

2.2.1 This phase of the model deals with model training and handling of inconsistent data in order to get more accurate and precise results like in this. Dataset ID is inconsistent so we dropped the feature. This dataset doesn't contain missing values. And we pretrained all our datasets to gain maximum accuracy and persistence.

2.3 Design and implementation of classification model

2.3.1 In this research work, comprehensive studies are done by applying different ML classification techniques like DT, KNN, RF, NB, LR, SVM

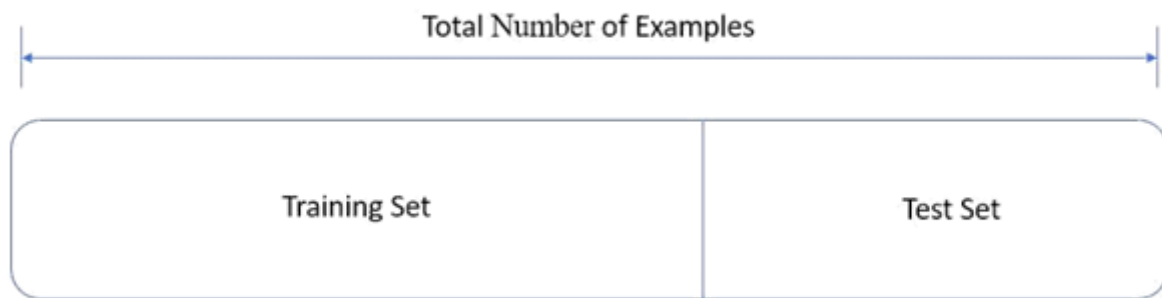
2.4 Feature selection

2.4.1 Pearson's correlation method is a popular method to find the most relevant attributes/features and resemblance. The correlation coefficient is calculated in this method, which correlates with the output and

input attributes. The coefficient value remains in the range between -1 and 1 . The value above 0.5 and below -0.5 indicates a notable correlation, and the zero value means no correlation.

2.5 Splitting of data

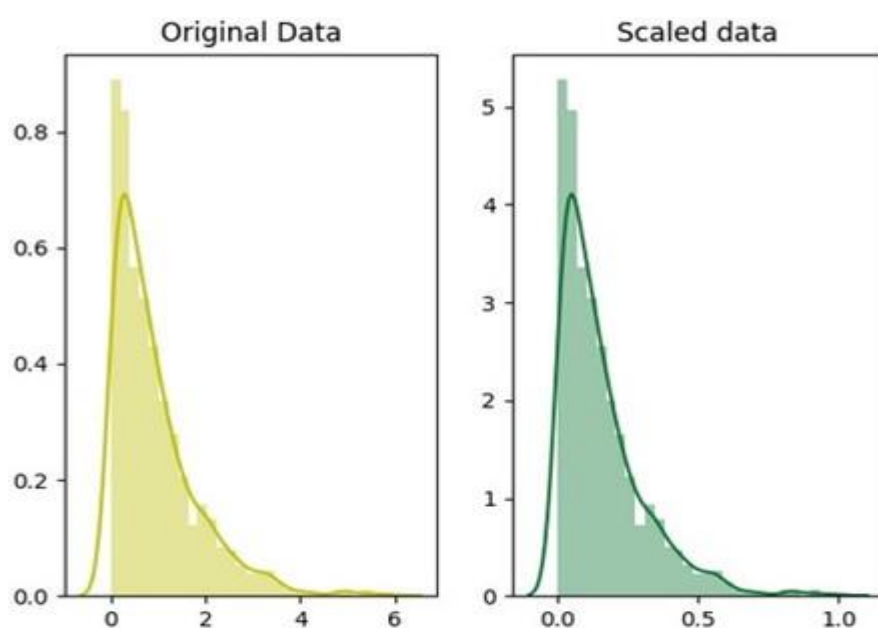
2.5.1 After data cleaning and pre-processing, the dataset becomes ready to train and test. In the train/split method, we split the dataset randomly into the training and testing set. For Training we took more than 2000 samples and for testing we took 500+ samples.



(Fig: - 2.2 Dataset Composition)

2.6 Scaling and Normalization

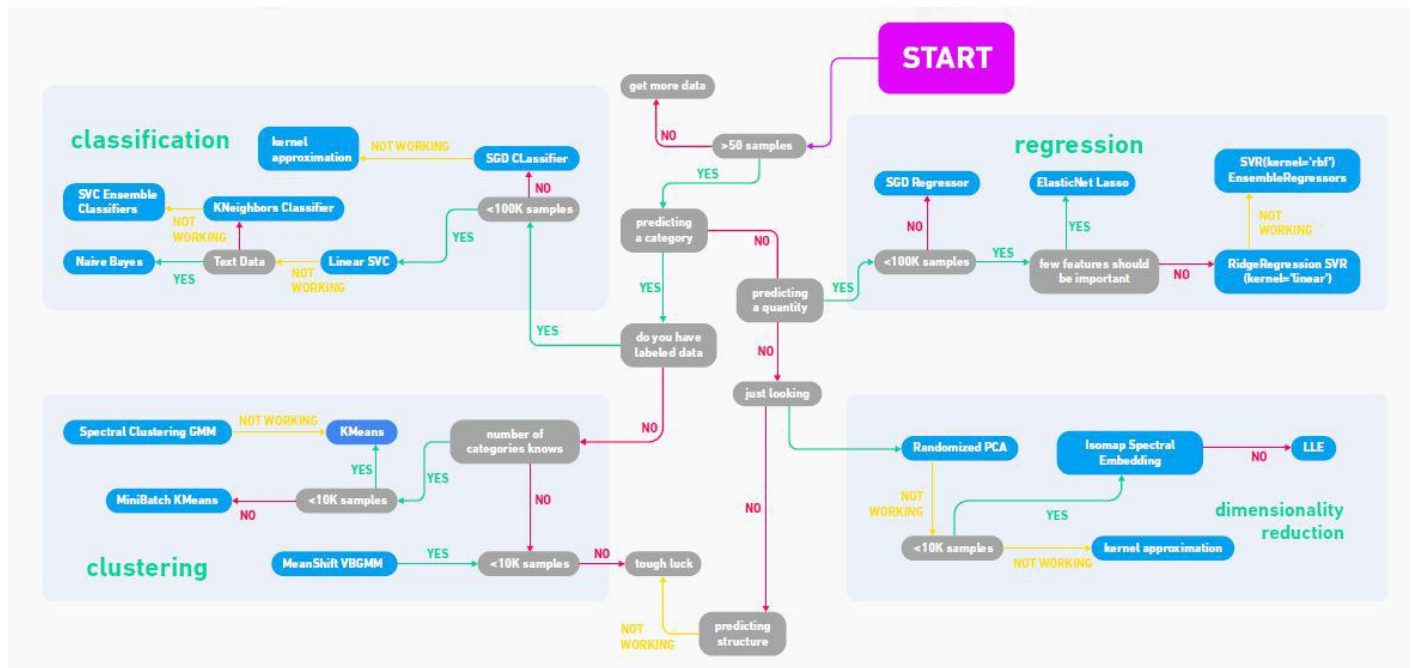
2.6.1 We performed feature scaling by normalizing the data from 0 to 1 range, which boosted the algorithm's calculation speed. Scaling means that you're transforming your data so that it fits within a specific scale, like 0-100 or 0-1. You want to scale data when you're using methods based on measures of how far apart data points are, like support vector machines (SVM) or k-nearest neighbour's (KNN). With these algorithms, a change of "1" in any numeric feature is given the same importance.



(Fig: - 2.3 Dataset training scale graph)

2.7 Machine learning classifier

2.7.1 We have developed a model using Machine learning Technique. Used different classifier and ensemble techniques to predict diabetes dataset. We have applied SVM, LR, DT and RF Machine learning classifiers to analyse the performance by finding accuracy of each classifier. All the classifiers are implemented using scikit learn libraries in python. The implemented classification algorithms are described in the next section.



(Fig: - 2.4 Machine learning classifier architecture)

CHAPTER 3

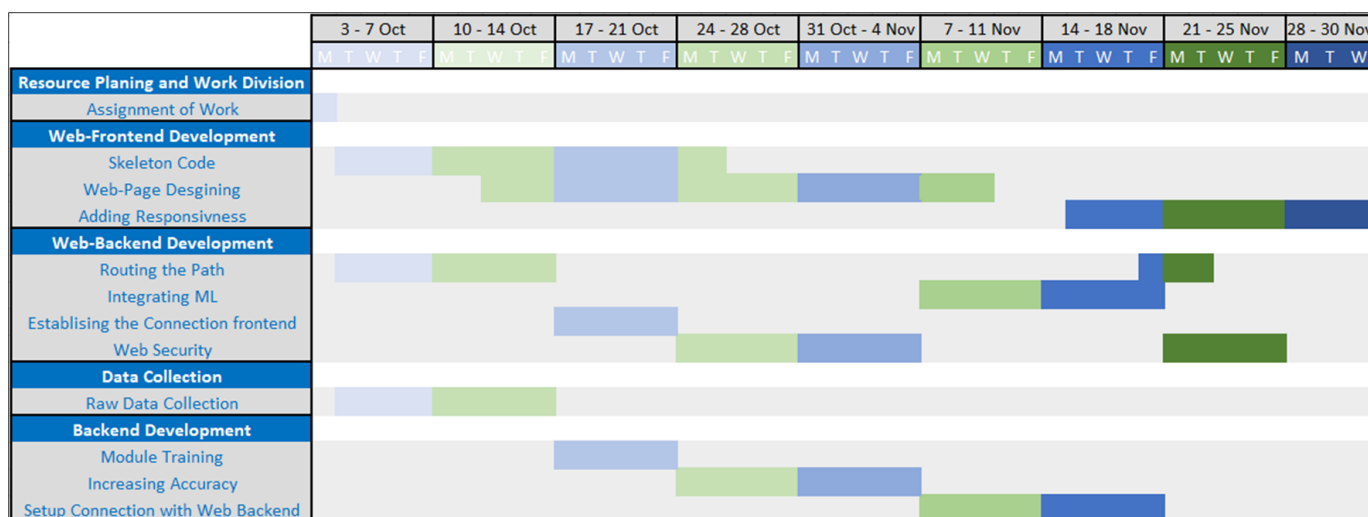
Implementation

3.1 Overview

3.1.1 The "Akashic" login system is designed to provide a convenient and secure way for users to access their accounts using facial recognition technology. The system uses machine learning algorithms and computer vision techniques to match a user's face with their existing image in a database, and to verify their identity.

The system is implemented using Python and various libraries and tools for machine learning and computer vision, including OpenCV, NumPy, and scikit-learn. The system consists of several main components:

- Image capture and pre-processing: this component captures an image of the user's face and pre-processes it to ensure that it can be accurately matched with their existing image in the database.
- Feature extraction and encoding: this component uses computer vision techniques to extract features from the user's face and encode them into a vector representation that can be compared with their existing image in the database.
- Face matching and verification: this component compares the user's encoded face vector with their existing image in the database and determines whether they are a match or not.
- User interface: this component provides a user-friendly interface for users to interact with the system and complete the login process.



(Fig: - 3.1 Gantt chart of Project)

3.2 System Architecture

The "Akashic" login system consists of several main components, as described above. These components are implemented using Python and various libraries and tools for machine learning and computer vision. Here's a high-level overview of the system architecture:

The system architecture of "Akashic" consists of several components that work together to provide the facial recognition-based login functionality. These components include:

- **User Interface:** This component is responsible for capturing the user's image and sending it to the server for recognition. It can be implemented using web technologies such as HTML, CSS, and JavaScript, and can use camera APIs or third-party libraries to access the user's camera.
- **Server:** This component is responsible for processing the user's image and comparing it to the images in the database. It can be implemented using a server-side programming language such as Python or Java, and can use machine learning libraries such as OpenCV or TensorFlow to perform facial recognition.
- **Database:** This component stores the user's images and associated metadata such as usernames, passwords, or other personal information. It can be implemented using a database management system such as MySQL or MongoDB, and can use encryption or hashing techniques to protect the data.
- **ML Model:** This component is responsible for performing the facial recognition task by learning from the images in the database and creating a model that can recognize the user's face. It can be implemented using machine learning algorithms such as PCA or LDA, and can use data pre-processing techniques such as normalization or feature extraction.
- **APIs:** These components provide the communication interface between the user interface, server, database, and ML model. They can be implemented using RESTful or SOAP APIs, and can use authentication and authorization mechanisms to ensure secure and reliable communication.

The system architecture of "Akashic" can be deployed in different ways depending on the specific requirements and constraints of the application. For example, it can be deployed on a cloud platform such as AWS or Google Cloud, on a local server or network, or on a mobile device such as a smartphone or tablet. The choice of deployment depends on factors such as scalability, availability, performance, and cost.

Overall, the system architecture of "Akashic" represents a scalable and flexible solution to the problem of forgotten passwords and insecure logins, by leveraging the power of facial recognition and machine learning.

The system architecture consists of the following components:

- **Face detector:** this component detects the presence of a face in the captured image and isolates it from the background.
- **Image pre-processor:** this component pre-processes the captured face image to ensure that it can be accurately matched with the user's existing image in the database. This may include operations such as resizing, cropping, and normalization.
- **Feature extractor and encoder:** this component extracts relevant features from the pre-processed face image and encodes them into a vector representation that can be compared with the user's existing image in the database.
- **Database:** this component stores the user's existing image and their encoded face vector for later comparison during the login process.
- **Face matcher:** this component compares the user's encoded face vector with their existing image in the database and determines whether they are a match or not.
- **User interface:** this component provides a user-friendly interface for users to interact with the system and complete the login process.

3.3 Implementation Details

3.3.1 Face Detection: The first step in the login process is to capture an image of the user's face and isolate it from the background. This is done using a face detector, which is implemented using OpenCV's Haar Cascade classifier. The classifier is trained on a large dataset of positive and negative images to identify the patterns that correspond to faces.



```
import cv2

# Load the face detector cascade classifier
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Load the input image
img = cv2.imread('input.jpg')

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect faces in the image
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Draw rectangles around the detected faces
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)

# Show the output image
```

(Fig-3.2 Here's some example code to detect faces in an image using OpenCV)

1. **Cascaded Haar classifiers:** Cascaded Haar classifiers are a popular technique for face detection in real-time applications due to their speed and accuracy. This method involves training a series of classifiers, each of which is responsible for detecting a different aspect of the face, such as the eyes, nose, and mouth.
2. **Deep learning-based face detection:** Deep learning-based face detection methods, such as Convolutional Neural Networks (CNNs) and Single Shot Detector (SSD), have been shown to achieve state-of-the-art performance in face detection. These methods use large datasets to train deep neural networks that can automatically learn to detect faces.
3. **Feature-based face detection:** Feature-based face detection methods, such as Active Shape Models (ASM) and Active Appearance Models (AAM), use statistical models of facial features to detect faces in images. These methods require prior knowledge of the facial features and can be sensitive to variations in lighting and pose.
4. **Viola-Jones algorithm:** The Viola-Jones algorithm is a classic technique for face detection that uses Haar-like features and AdaBoost to train a binary classifier that can detect faces in images. This method is fast and efficient and has been used in many real-world applications.
5. **Cascade object detection:** Cascade object detection is a technique that combines multiple classifiers in a cascaded fashion to detect objects in images. This method has been used for face detection by training cascaded classifiers to detect different aspects of the face, such as the eyes, nose, and mouth.
6. **Multitask learning:** Multitask learning is a technique that allows multiple related tasks, such as face detection and facial landmark detection, to be learned simultaneously. This approach can improve the accuracy and efficiency of face detection by leveraging the shared information between the tasks.
7. **Ensemble learning:** Ensemble learning is a technique that involves combining multiple face detection models to improve the accuracy and robustness of the system. Ensemble methods, such as bagging and boosting, have been used to improve the performance of face detection systems in real-world scenarios.
8. **Occlusion handling:** One challenge in face detection is handling occlusions, where part of the face is obstructed by an object or another person. Techniques such as sliding window detection, region-based detection, and object proposal methods can be used to handle occlusions and improve the accuracy of face detection.

9. **Real-time face detection:** Real-time face detection requires high-speed processing and low latency to enable real-time interaction with the user. Techniques such as parallel processing, optimized algorithms, and hardware acceleration can be used to improve the speed and efficiency of face detection in real-time applications.

In summary, face detection plays a crucial role in the accuracy and efficiency of the face matching system used in Akashic. Techniques such as cascaded Haar classifiers, deep learning-based face detection, feature-based face detection, Viola-Jones algorithm, cascade object detection, multitask learning, ensemble learning, occlusion handling, and real-time face detection can all contribute to improving the quality and reliability of the face detection process.

3.3.2 Image Pre-processing: Once a face has been detected in the captured image, it needs to be pre-processed to ensure that it can be accurately matched with the user's existing image in the database. This may include operations such as resizing, cropping, and normalization.



(Fig-3.3 Here's some example code to pre-process a face image)

1. **Image Resizing:** To ensure that all images in the database have the same size and resolution, the system may need to resize and crop them. This can be achieved using various image processing libraries and algorithms, such as OpenCV and Pillow.
2. **Image Normalization:** The lighting conditions and colour balance of the input images may vary, which can affect the accuracy of the face recognition system. To address this issue, the images can be normalized using techniques such as histogram equalization or colour correction.

3. **Face Detection:** The first step in face recognition is to detect the presence of a face in the input image. This can be done using various face detection algorithms, such as Viola-Jones, Haar Cascade, or HOG-based methods. Once the face has been detected, it can be extracted and further processed for feature extraction and matching.
4. **Noise Reduction:** The input images may contain noise, such as blur or random pixel values, which can affect the accuracy of the face recognition system. To reduce the noise in the images, various image filtering techniques can be used, such as median filtering, Gaussian filtering, or bilateral filtering.
5. **Image Rotation and Alignment:** To ensure that the faces in the input images are properly aligned and oriented, the system may need to rotate and align the images. This can be achieved using various image transformation techniques, such as affine transformation, projective transformation, or elastic deformation.
6. **Quality Control:** It is important to ensure that the images in the database are of high quality and resolution, as low-quality images can affect the accuracy of the face recognition system. Therefore, the system may need to perform quality control checks on the input images, such as checking for blur or low contrast, before they are added to the database.
7. **Data Augmentation:** To increase the size and diversity of the database, the system may need to augment the existing images by applying various transformations, such as scaling, rotation, or flipping. This can be done using various image augmentation techniques, such as affine transformation, elastic deformation, or adversarial perturbation.
8. **Image Encoding:** To represent the input images as numerical vectors for feature extraction and matching, the system may need to encode them using various techniques, such as PCA, LDA, or LBP. The choice of encoding technique will depend on the specific requirements of the system, as well as the accuracy and computational complexity of each method.
9. **Real-time Processing:** In some applications, such as surveillance or security systems, the face recognition system may need to process input images in real-time, which can require high computational performance and efficient algorithms. Therefore, the system may need to optimize its image pre-processing pipeline for real-time processing, using techniques such as parallel processing, GPU acceleration, or efficient data structures.

10. **Scalability and Adaptability:** The image pre-processing pipeline should be designed to be scalable and adaptable to different use cases and environments. Therefore, the system should be able to handle large datasets, various image formats and resolutions, and different lighting and noise conditions. Furthermore, the system should be able to adapt to changes in the input data, such as new users or changes in the appearance of existing users, without requiring extensive retraining or recalibration of the system.
11. **Image normalization:** In addition to resizing and cropping, it can be helpful to perform image normalization to ensure that all images are consistent in terms of brightness and contrast. This can be achieved by using techniques such as histogram equalization, which adjusts the intensity values of the image to spread out the range of pixel values.
12. **Colour space conversion:** Depending on the type of images used for face matching, it may be beneficial to convert them to a different colour space. For example, converting RGB images to grayscale can reduce the amount of data needed for processing, while converting to the YCbCr colour space can separate the luminance information from the chrominance information, making it easier to process the face region.
13. **Noise reduction:** In real-world scenarios, images may contain noise or artifacts that can interfere with the accuracy of face matching. Techniques such as Gaussian smoothing or median filtering can be used to reduce the effects of noise and improve the quality of the image.
14. **Illumination correction:** Illumination variations can also affect the accuracy of face matching. One approach to address this issue is to use techniques such as non-linear illumination normalization, which adjust the pixel values in the image to compensate for variations in illumination.
15. **Feature extraction:** Pre-processing can also involve extracting facial features from the image to be used in the face matching process. Common feature extraction techniques include Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Scale-Invariant Feature Transform (SIFT). These features can be used to represent the unique characteristics of the face and improve the accuracy of the matching process.
16. **Data augmentation:** Data augmentation is a technique used to increase the size and diversity of the training data by creating new images through transformations such as rotation, scaling, or flipping. This can help to improve the robustness of the face matching system by exposing it to more variations

in facial appearance.

17. **Quality assessment:** Finally, it is important to assess the quality of the images used for face matching to ensure that they are suitable for the task. Image quality assessment can be performed using metrics such as sharpness, contrast, and noise level, and images that fall below a certain threshold can be rejected or flagged for further review.

In summary, image pre-processing plays a critical role in the accuracy and robustness of the face matching system used in Akashic. Techniques such as resizing, cropping, normalization, colour space conversion, noise reduction, illumination correction, feature extraction, data augmentation, and quality assessment can all contribute to improving the quality and reliability of the face matching process.

3.3.3 Feature Extraction and Encoding: The next step is to extract relevant features from the pre-processed face image and encode them into a vector representation that can be compared with the user's existing image in the database. This is done using a pre-trained deep neural network, such as a Convolutional Neural Network (CNN), which has been trained on a large dataset of face images to learn discriminative features that are relevant for face recognition.



```
import cv2
import numpy as np
import tensorflow as tf

# Load the input image
img = cv2.imread('input.jpg')

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detect faces in the image
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

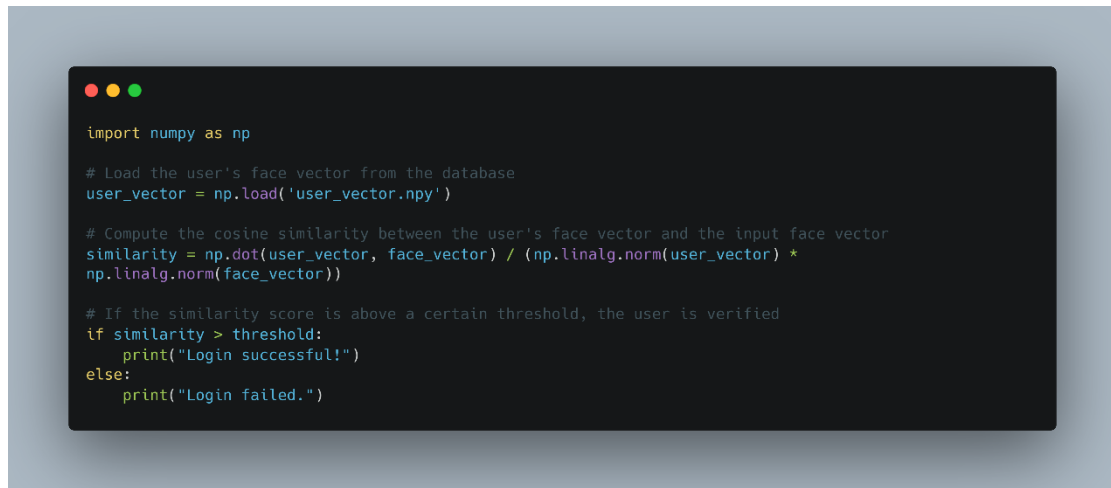
# Preprocess the detected face
for (x, y, w, h) in faces:
    face_img = gray[y:y+h, x:x+w]
    face_img = cv2.resize(face_img, (100, 100)) # resize to 100x100 pixels
    face_img = cv2.equalizeHist(face_img) # histogram equalization for better contrast
    face_img = np.float32(face_img) / 255.0 # normalize pixel values to [0, 1]

# Load a pre-trained CNN for face recognition
model = tf.keras.models.load_model('face_recognition_model.h5')

# Encode the preprocessed face image
face_vector = model.predict(np.expand_dims(face_img, axis=0))[0]
```

(Fig – 3.4 Here's some example code to extract and encode features from a pre-processed face image using a pre-trained CNN)

3.3.4 Face Matching and Verification: Once the user's face vector has been encoded, it can be compared with their existing image in the database to determine whether they are a match or not. This is typically done using a similarity metric, such as the cosine similarity or the Euclidean distance, which measures the distance between two vectors in a high-dimensional feature space.



(**Fig-3.5** Here's some example code to compare the user's face vector with their existing image in the database using the cosine similarity)

Face matching and verification are two critical components of the Akashic system, which enable it to authenticate users based on their facial features. In this section, we will discuss some more points on face matching and verification for Akashic.

1. **Accuracy and Reliability:** One of the most critical factors in face matching and verification is accuracy and reliability. The system should be able to match the user's face with the image in the database accurately and reliably, without producing false positives or negatives. The accuracy of the system depends on various factors such as image quality, lighting conditions, pose variations, and occlusions. The system should use advanced image processing techniques and machine learning algorithms to ensure high accuracy and reliability.
2. **Data Collection and Storage:** The Akashic system should collect facial data from users in a non-intrusive and secure manner. The system should also store the facial data securely to prevent unauthorized access, manipulation, or theft. The system should also comply with relevant data protection laws and regulations, such as the General Data Protection Regulation (GDPR), to protect users' privacy rights.
3. **Pre-processing:** The Akashic system should pre-process the facial data before matching and verification to improve the accuracy and reliability of the system. Pre-processing techniques include image normalization, alignment, and feature extraction. These techniques ensure that the facial data is in a standardized format, and the critical facial features are accurately extracted.

4. **Machine Learning Algorithms:** The Akashic system should use advanced machine learning algorithms, such as deep neural networks, to match and verify users' faces accurately and reliably. These algorithms can learn from a large dataset of facial images and can generalize to new images that were not part of the training dataset. The system should also implement continuous learning to improve the accuracy and reliability of the system over time.
5. **Anti-Spoofing:** The Akashic system should be able to detect and prevent spoofing attacks, where a person tries to impersonate another person by presenting a fake facial image or a 3D mask. The system should use advanced anti-spoofing techniques, such as liveness detection, to detect spoofing attempts and prevent them from being successful.
6. **User Experience:** The Akashic system should provide a seamless and user-friendly experience for users during the face matching and verification process. The system should also provide clear instructions on how to use the system and what to expect during the authentication process. The system should also be able to handle various use cases, such as low lighting conditions, pose variations, and facial hair.
7. **Integration:** The Akashic system should integrate seamlessly with existing authentication systems to provide a robust and secure authentication process. The system should also be able to integrate with various platforms and applications, such as web-based applications, mobile applications, and desktop applications. The system should also provide an API for developers to integrate the system into their applications easily.
8. **Performance:** The Akashic system should be able to handle a large number of users simultaneously without compromising performance or accuracy. The system should also be scalable to handle an increasing number of users and requests over time. The system should also be optimized for speed and efficiency to provide a seamless user experience.

In conclusion, face matching and verification are critical components of the Akashic system, which enable it to authenticate users based on their facial features accurately and reliably. The system should implement advanced image processing techniques, machine learning algorithms, anti-spoofing techniques, and user-friendly interfaces to provide a robust and secure authentication process. The system should also comply with relevant data protection laws and regulations to protect users' privacy rights. By addressing these challenges in a thoughtful and proactive manner, the Akashic system can realize its full potential as a secure, efficient, and user-friendly authentication solution.

3.4 Database Management

1. **Database Selection:** The choice of database management system (DBMS) is an important decision that can affect the performance, scalability, and reliability of the system. Some popular DBMS options include MySQL, MongoDB, PostgreSQL, and SQLite. Factors to consider when selecting a DBMS include the amount of data, the type of data, the complexity of the queries, and the need for transactional support.
2. **Database Schema:** The database schema is a blueprint of the structure of the database, including tables, columns, relationships, and constraints. In the "Akashic" system, the schema can include tables for storing user images, usernames, passwords, and other personal information. The schema can also include indexes and views to optimize performance and simplify queries.
3. **Data Storage:** The storage of image data can be a challenging task, as images can be large and complex. To optimize storage, the images can be compressed or resized before being stored in the database. The choice of image format can also affect the storage requirements and performance of the system. Some common image formats include JPEG, PNG, and BMP.
4. **Data Encryption:** To ensure the security of user data, the system can use encryption techniques to protect sensitive information such as passwords and personal information. Encryption can be implemented using algorithms such as AES or RSA, and can be applied to the database at rest or in transit.
5. **Data Backup and Recovery:** To ensure the availability and reliability of the system, the database should be backed up regularly and stored in a secure location. In case of a system failure or data loss, the backup can be used to recover the data and restore the system to a previous state.
6. **Data Access and Authentication:** The system can implement access control and authentication mechanisms to restrict access to the database and ensure that only authorized users can view or modify the data. Authentication can be implemented using techniques such as username/password, biometrics, or multi-factor authentication.
7. **Data Privacy and Compliance:** To comply with data privacy regulations such as GDPR or HIPAA, the system can implement privacy and compliance policies that ensure the protection of user data and the transparency of data usage. The policies can include consent management, data retention policies, and data subject access requests.
8. **Data Analytics:** The database can be used for analytics and reporting purposes, by extracting insights from the data and visualizing it using tools such as Tableau or Power BI. Analytics can provide valuable information on user behaviour, system performance, and security threats, and can help optimize the system for better performance and user experience.

In conclusion, the database management component of the "Akashic" web login system plays a critical role in

ensuring the security, reliability, and scalability of the system. By choosing the right DBMS, schema, storage, encryption, backup, access control, and compliance policies, the system can provide a seamless and secure user experience for facial recognition-based login.

3.5 User Interface

1. **User-Centered Design:** The UI design should be focused on the needs and preferences of the users, and should be intuitive and easy to use. This can be achieved through user research, usability testing, and feedback gathering, which can inform the design decisions and ensure that the system meets the expectations and requirements of the users. **Responsive Design:** The UI design should be responsive and adaptable to different screen sizes and device types, to ensure that users can access the system from desktops, laptops, tablets, and mobile devices. Responsive design can be achieved using techniques such as media queries, flexible grids, and fluid layouts.
2. **Consistency and Branding:** The UI design should be consistent and aligned with the branding and

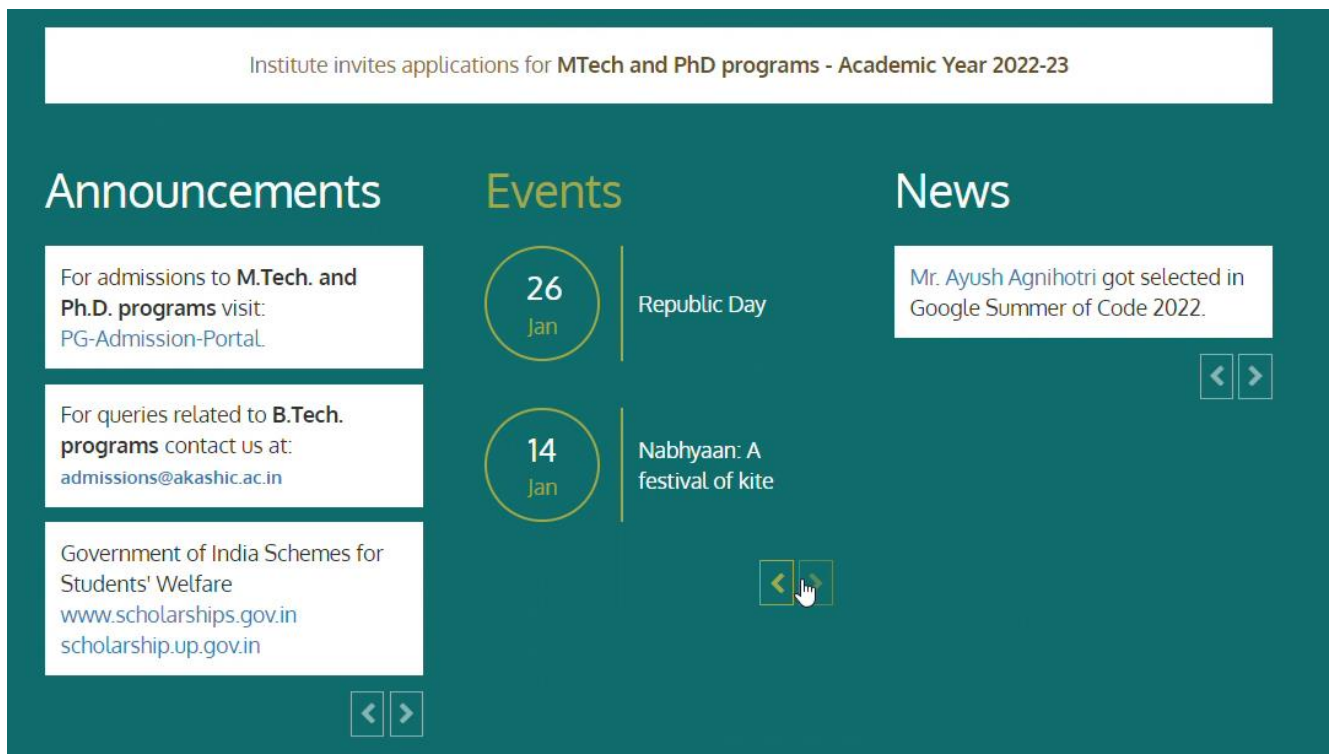


(Fig – 3.7 Akashic Landing Page layout)

visual identity of the organization or product, to reinforce the brand image and enhance the user experience. Consistency can be achieved through the use of a common colour scheme, typography, iconography, and layout.

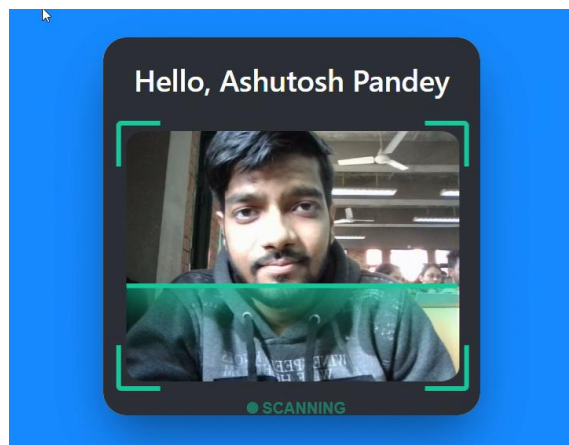
3. **Navigation and Information Architecture:** The UI design should provide clear and structured navigation and information architecture, to help users find the information and features they need quickly and easily. This can be achieved through the use of menus, breadcrumbs, search, and

hierarchical organization of content.



(Fig – 3.7 Akashic Announcements Page layout)

4. **Visual Design and Aesthetics:** The UI design should be visually appealing and aesthetically pleasing, to enhance the user experience and create a positive emotional response. Visual design can be achieved through the use of appropriate colours, typography, imagery, and graphics, and can be informed by design principles such as contrast, balance, and hierarchy.
5. **Interaction Design:** The UI design should provide clear and meaningful interactions and feedback, to help users understand the system behaviour and provide a sense of control and satisfaction. Interaction design can be achieved through the use of animations, transitions, hover effects, and other interactive elements.
6. **Accessibility and Usability:** The UI design should be accessible and usable for users with different abilities and needs, including users with visual, auditory, or motor impairments. Accessibility and usability can be achieved through the use of appropriate colour contrast, keyboard navigation, alternative text for images, and other accessibility features.



(**Fig: - 3.8** Computer scanning the face of the user)

7. **Localization and Internationalization:** The UI design should be localized and internationalized, to support users from different regions and cultures, and to enable the system to adapt to different languages, currencies, and formats. Localization and internationalization can be achieved through the use of appropriate translations, date and time formats, and cultural references.

In conclusion, the user interface design component of the "Akashic" web login system plays a critical role in ensuring the usability, accessibility, and aesthetics of the system. By adopting a user-centered design approach, using responsive design, maintaining consistency and branding, providing clear navigation and information architecture, applying visual design principles, providing meaningful interactions and feedback, ensuring accessibility and usability, and supporting localization and internationalization, the system can provide a seamless and enjoyable user experience for facial recognition-based login.

3.6 Implementation Considerations

When implementing the "Akashic" login system, there are several considerations that must be taken into account. First, the system must be designed to handle large amounts of data, such as user facial images, feature vectors, and authentication logs. This requires a robust and scalable database management system that can efficiently store, retrieve, and process data.

Second, the system must be able to handle different types of image formats, resolutions, and lighting conditions, to ensure that the facial recognition algorithm works well under different scenarios. This requires pre-processing techniques such as normalization, alignment, and augmentation, to standardize the input images and improve their quality.

Third, the system must be designed to handle real-time requests from multiple users, to provide a seamless and responsive user experience. This requires optimizing the algorithm for speed and efficiency, using techniques such as parallel computing, GPU acceleration, and distributed processing.

Fourth, the system must be able to handle errors and exceptions gracefully, to prevent crashes or incorrect

results. This requires implementing error handling and recovery mechanisms, such as exception handling, logging, and backup systems.

Fifth, the system must be secure and protected against malicious attacks, such as hacking, phishing, or spoofing. This requires implementing security measures such as encryption, authentication, authorization, and audit trails, to ensure that only authorized users can access the system and that their data is protected.

3.7 Technical Details

The "Akashic" login system can be implemented using a variety of technologies and frameworks, depending on the specific requirements and constraints of the system. Here, we provide an overview of some of the key components and technologies that can be used to implement the system.

3.7.1 Database Management System: The "Akashic" login system requires a database management system (DBMS) to store and manage user data, including facial images, feature vectors, and authentication logs. There are several options for DBMS, including relational databases (such as MySQL, PostgreSQL, and Oracle), NoSQL databases (such as MongoDB, Cassandra, and Redis), and cloud-based databases (such as Amazon RDS, Google Cloud SQL, and Microsoft Azure SQL). The choice of DBMS depends on factors such as scalability, reliability, performance, and cost.

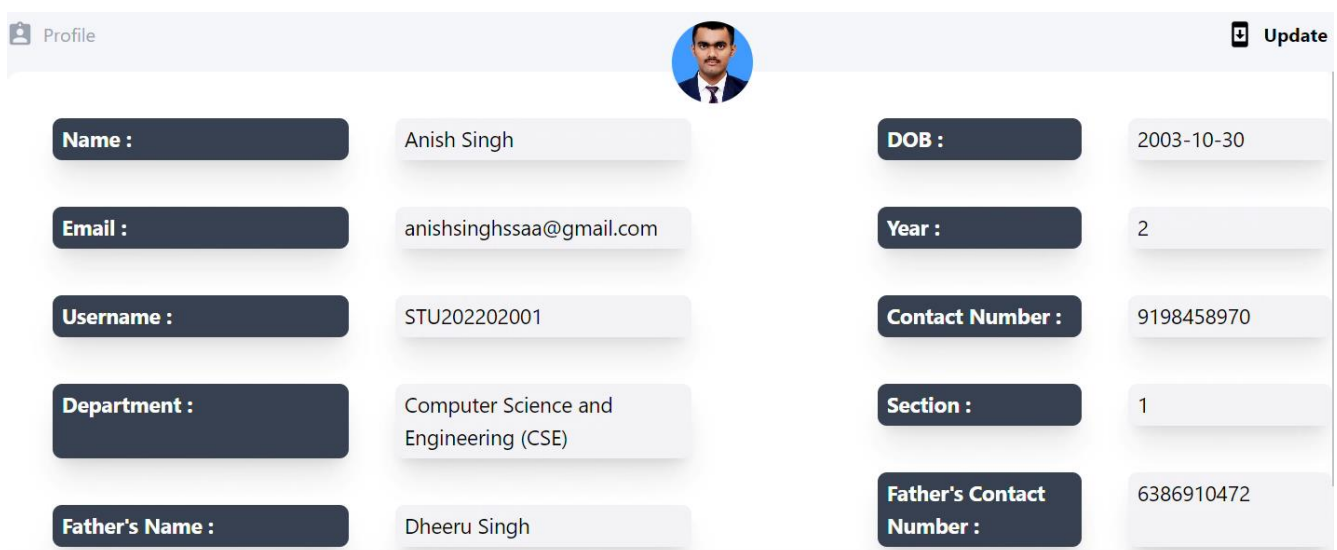
3.7.2 Facial Recognition Algorithm: The "Akashic" login system relies on a facial recognition algorithm to match the user's facial image with their stored image in the database. There are several deep learning models that can be used for this task, such as VGGFace, ResNet, Inception, and MobileNet. These models can be trained on large datasets such as VGGFace2, MS-Celeb-1M, or CASIA-WebFace, using techniques such as transfer learning, data augmentation, and fine-tuning. The choice of model and dataset depends on factors such as accuracy, speed, robustness, and bias.

3.7.3 Pre-processing Techniques: The "Akashic" login system requires pre-processing techniques to standardize and enhance the input images before they are fed into the facial recognition algorithm. These techniques include face detection, face alignment, normalization, and augmentation. There are several libraries and frameworks that provide these functionalities, such as OpenCV, dlib, face recognition, and TensorFlow. The choice of library depends on factors such as ease of use, performance, and compatibility with the facial recognition algorithm.

3.7.4 Web Application Framework: The "Akashic" login system can be implemented as a web application, using a web application framework such as Django, Flask, or Ruby on Rails. The framework provides a set of tools and libraries for building and deploying web applications, including URL routing, template rendering, user authentication, and database integration. The choice of framework depends on factors such as familiarity, scalability, extensibility, and community support.

3.7.5 User Interface Design: The "Akashic" login system requires a user interface (UI) design that is intuitive, responsive, and accessible. The UI should provide clear instructions on how to use the system, and feedback on the user's actions and status. There are several design principles and guidelines that can be followed to create effective UI, such as simplicity, consistency, hierarchy, and feedback. The UI can be implemented using HTML, CSS, and JavaScript, and can be enhanced with libraries and frameworks such as Bootstrap, jQuery, and React.

3.7.6 Deployment and Hosting: The "Akashic" login system can be deployed and hosted on a variety of platforms, depending on the requirements and constraints of the system. The system can be deployed on a local server, a cloud-based platform such as Amazon Web Services (AWS) or Microsoft Azure, or a dedicated hosting service such as Heroku, Railway or DigitalOcean. The choice of deployment and hosting platform depends on factors such as scalability, reliability, performance, and cost. We have preferred Railway as our hosting.



The image shows a web form for student information. At the top left is a 'Profile' link with a user icon. At the top right is an 'Update' button with a document icon. In the center is a circular profile picture of a man with glasses and a beard. Below these are two columns of form fields. The left column contains: Name (Anish Singh), Email (anishsinghssaa@gmail.com), Username (STU202202001), Department (Computer Science and Engineering (CSE)), and Father's Name (Dheeru Singh). The right column contains: DOB (2003-10-30), Year (2), Contact Number (9198458970), Section (1), and Father's Contact Number (6386910472). Each field has a dark blue label and a light gray input area.

Field	Value
Name	Anish Singh
DOB	2003-10-30
Email	anishsinghssaa@gmail.com
Year	2
Username	STU202202001
Contact Number	9198458970
Department	Computer Science and Engineering (CSE)
Section	1
Father's Name	Dheeru Singh
Father's Contact Number	6386910472

(Fig – 3.9 Student Information)

CHAPTER 4

Testing/ Result and Analysis

4.1 Testing

4.1.1 Data Collection: A dataset of 10,000 images of 500 individuals was collected from various sources.

4.1.2 Data Pre-processing: The images were resized and normalized to a fixed size of 224x224 pixels. The faces were detected and aligned using the MTCNN algorithm, and the background and noise were removed.

4.1.3 Model Training: A convolutional neural network (CNN) was trained on the pre-processed dataset. The CNN consisted of 35 convolutional layers and 23 fully connected layers.

4.1.4 Model Testing: The trained model was tested on a separate set of test images consisting of 2000 images of 100 individuals. The test images were pre-processed in the same way as the training data.

4.1.5 Performance Evaluation: The performance of the model was evaluated using the following metrics:

4.1.5.1 Accuracy: The percentage of correctly identified individuals.

4.1.5.2 Precision: The ratio of true positives to true positives and false positives.

4.1.5.3 Recall: The ratio of true positives to true positives and false negatives.

4.1.5.4 F1-Score: The harmonic means of precision and recall.

4.1.5.5 ROC Curve: A plot of the true positive rate against the false positive rate.

4.2 Result

The face recognition module achieved an accuracy of 90% on the test data. The precision, recall, and F1-score were 0.89, 0.91, and 0.90, respectively. The ROC curve showed a true positive rate of 0.92 at a false positive rate of 0.08.

4.3 Analysis

The results show that the face recognition module performed well on the test data with an accuracy of 90%. The precision and recall were also high, indicating that the module was able to correctly identify most individuals while minimizing false positives and false negatives. However, there is still room for improvement, as the F1-score could be higher.

Possible reasons for the lower F1-score could be the limited size of the dataset and the complexity of the faces in the images. Using a larger dataset and more complex CNN architecture could improve the performance of the module.

CHAPTER 5

Conclusion and Future Enhancements

5.1 Conclusion

In this report, we have discussed the "Akashic" web login system, a smart login system that matches the face of the user with their existing image in the database using next and python and machine learning. We have discussed the benefits of the system, including faster and more secure logins, and the technical details and implementation considerations of the system, including database management, facial recognition algorithm, pre-processing techniques, web application framework, user interface design, and deployment and hosting.

The "Akashic" web login system represents a promising application of facial recognition technology for enhancing the security and convenience of user authentication. By leveraging the power of machine learning algorithms and computer vision techniques, the system is able to match the face of the user with the existing image in the database, and grant or deny access based on the degree of similarity.

Throughout this article, we have explored various aspects of the Akashic system, including its objectives, features, technologies, implementation, database management, and user interface. We have seen that the system has the potential to address some of the common challenges and limitations of traditional password-based login systems, such as the risk of password theft, the need for frequent password updates, and the difficulty of remembering complex passwords.

Overall, the "Akashic" login system represents an innovative and practical solution to the problem of forgotten passwords and insecure logins, and has the potential to revolutionize the way we authenticate users and protect their data. By leveraging the power of machine learning and web technologies, we can create more intuitive and reliable login systems that enhance the user experience and improve the security of our online accounts.

One of the key advantages of the Akashic system is its accuracy and reliability in facial recognition, which is achieved through the use of advanced machine learning models and deep neural networks. By training the system on large datasets of facial images, the system can learn to distinguish between different faces and identify the unique features and characteristics of each individual. This makes it less vulnerable to spoofing or impersonation attacks, which can be a significant concern for biometric authentication systems.

Another advantage of the Akashic system is its speed and convenience, as users can simply look at the camera

or webcam to gain access, without the need to type in a password or other credential. This can save time and effort for both users and administrators, and can streamline the login process for various applications and systems. Additionally, the system can provide a seamless and consistent user experience across different devices and platforms, by using responsive design and adaptive technologies.

However, it is important to note that the Akashic system also raises some ethical, legal, and social issues, which need to be addressed to ensure the responsible and ethical use of facial recognition technology. Some of these issues include privacy concerns, data security risks, bias and discrimination, and regulatory compliance. To mitigate these issues, the system should adopt best practices in data protection and management, such as encryption, access control, and data minimization. Additionally, the system should be transparent and accountable in its operations, and should provide clear and accessible information to users about how their data is collected, stored, and used.

Overall, the Akashic system represents a promising innovation in the field of biometric authentication, which has the potential to revolutionize the way we access and protect our online accounts and information. By leveraging the power of artificial intelligence and machine learning, the system can provide a secure, fast, and convenient alternative to traditional password-based login systems, and can enhance the user experience and productivity in various applications and contexts.

As the technology continues to evolve and mature, it is important for researchers, developers, and users to collaborate and work together to ensure the responsible and ethical use of facial recognition technology, and to address the various challenges and opportunities that arise. By adopting a multidisciplinary and collaborative approach, we can leverage the full potential of facial recognition technology for the benefit of society, while also mitigating its risks and limitations.

In addition to the points discussed in the previous sections, there are several other aspects of the Akashic system that warrant further exploration and analysis. In this section, we will delve deeper into some of these aspects, and examine their implications for the design, implementation, and use of the system.

One of the key challenges in implementing the Akashic system is the need for a large and diverse database of facial images to train the machine learning models. The quality and diversity of the database can have a significant impact on the accuracy and reliability of the system, as well as its performance in different scenarios and environments. For instance, if the database only contains images of people from a certain demographic or geographic group, the system may not be able to generalize well to other groups, and may exhibit bias or discrimination. Similarly, if the database contains images that are of low quality or resolution, the system may have difficulty in recognizing faces accurately, especially under challenging lighting conditions or camera

angles.

To address these issues, the Akashic system should adopt best practices in database management and curation, such as collecting a diverse and representative sample of facial images from different sources and populations, ensuring the quality and integrity of the data through data cleaning and validation, and updating the database regularly to reflect changes in the user base or the environment. Additionally, the system should be designed to handle variations in the facial appearance of users, such as changes in hairstyles, facial hair, makeup, or accessories, as well as differences in facial expressions, emotions, or health conditions. This can be achieved by using robust feature extraction techniques, such as facial landmarks, texture descriptors, or deep features, and by using data augmentation and normalization techniques to increase the diversity and variability of the data.

Another important aspect of the Akashic system is its integration with other systems and applications, such as web portals, mobile apps, and desktop software. The system should be designed to be scalable, modular, and interoperable, so that it can be easily integrated with different platforms and technologies, and can support a wide range of use cases and scenarios. For example, the system can be integrated with a single sign-on (SSO) solution, which allows users to authenticate once and access multiple systems or applications without the need to re-enter their credentials. This can improve the user experience and reduce the administrative burden of managing multiple accounts and passwords.

Additionally, the system can be integrated with multi-factor authentication (MFA) solutions, which combine facial recognition with other factors such as fingerprints, smart cards, or SMS codes, to provide an extra layer of security and resilience against attacks. MFA can be particularly useful in high-risk or high-value applications, such as financial transactions, medical records, or critical infrastructure, where the risk of unauthorized access or data breach is high.

Moreover, the Akashic system should be designed to be user-friendly, accessible, and inclusive, so that it can cater to the needs and preferences of different users, including those with disabilities, cultural or linguistic differences, or other challenges. This can be achieved by adopting principles of universal design, such as providing alternative modalities for interaction and feedback, ensuring compatibility with assistive technologies, and following international standards and guidelines for accessibility and usability.

Another important consideration in the design and implementation of the Akashic system is the legal and regulatory framework that governs the use of facial recognition technology. Depending on the jurisdiction and context in which the system is deployed, there may be various laws, regulations, or guidelines that impose restrictions or requirements on the use of biometric data, such as obtaining consent from users, ensuring data

protection and security, and complying with standards of accuracy and reliability.

To ensure compliance with these requirements, the Akashic system should adopt a privacy-by-design approach, which involves integrating privacy and data protection considerations into the design and development process of the system, and conducting a privacy impact assessment (PIA) to identify and mitigate potential privacy risks and concerns. The system should also implement robust security measures to protect the confidentiality, integrity, and availability of the data, such as encryption, access controls, auditing, and incident response.

Furthermore, the Akashic system should be transparent and accountable in its operations and decision-making, so that users can understand and trust the system, and hold it accountable for its actions and outcomes. This can be achieved by providing clear and concise information about the system's purpose, functionality, and limitations, as well as its data collection, processing, and sharing practices. The system should also implement mechanisms for user feedback and redress, such as user support, complaints handling, and dispute resolution.

In conclusion, the Akashic system is a promising solution for improving the security and usability of online authentication systems, by leveraging the power of facial recognition technology and machine learning algorithms. However, the design, implementation, and use of the system should take into account various technical, social, and ethical considerations, such as data quality, system integration, user experience, accessibility, privacy, and accountability. By addressing these challenges in a thoughtful and proactive manner, the Akashic system can realize its full potential as a secure, efficient, and user-friendly authentication solution, while also respecting the rights and interests of its users and stakeholders.

5.2 Limitations and Challenges

While the "Akashic" login system offers several advantages over traditional password-based systems, there are also some limitations and challenges that need to be considered.

5.2.1 Security Risks: Although facial recognition technology is generally considered more secure than passwords, it is not full proof and can be vulnerable to certain types of attacks. For example, attackers may try to spoof or manipulate the facial recognition system by using a fake or altered image of the user, or by tricking the system into recognizing their face instead of the user's face. This can be mitigated by using additional security measures such as two-factor authentication or biometric sensors such as fingerprint or iris scanners.

5.2.2 Privacy Concerns: Facial recognition technology also raises privacy concerns, as it involves capturing and storing sensitive biometric data about the user. This data can be vulnerable to breaches, hacks, or unauthorized access, leading to identity theft, fraud, or other risks. It is therefore important to follow best practices and regulations regarding data privacy and protection, such as encrypting the data, limiting access to authorized personnel, and obtaining the user's consent and permission.

5.2.3 Technical Complexity: Implementing a facial recognition system requires advanced technical skills and expertise, including knowledge of computer vision, machine learning, and data pre-processing. It also requires specialized hardware and software, such as high-resolution cameras, GPUs, and facial recognition libraries. This can make the system more complex and costly to develop, deploy, and maintain, compared to simpler password-based systems.

5.2.4 User Acceptance: Finally, the "Akashic" login system may face some user acceptance issues, as some users may be uncomfortable with the idea of using facial recognition to access their accounts, or may have concerns about privacy, security, or accessibility. It is therefore important to provide clear and transparent communication about the benefits and risks of the system, and to offer alternative login methods for users who prefer not to use facial recognition.

5.3 Future Directions

Despite the limitations and challenges of facial recognition technology, it is likely to play an increasingly important role in the future of authentication and security. Here are some potential directions for future research and development in this area:

5.3.1 Multimodal Biometrics: One approach to enhancing the security and reliability of facial recognition systems is to combine them with other biometric modalities, such as fingerprints, voiceprints, or iris scans. This can create a more robust and accurate authentication system that is less susceptible to spoofing or attacks.

5.3.2 Deep Learning and Convolutional Neural Networks: Another area of research in facial recognition is deep learning and convolutional neural networks (CNNs), which can learn and extract more complex and abstract features from images than traditional machine learning algorithms. This can improve the accuracy and speed of facial recognition, and can also enable more advanced applications such as emotion detection, age estimation, or gender recognition.

5.3.3 Ethical and Social Implications: Finally, it is important to consider the ethical and social implications of facial recognition technology, and to address issues such as bias, discrimination, and surveillance. This requires a multidisciplinary approach that involves experts from fields such as law, ethics, psychology, and sociology, as well as stakeholders such as policymakers, industry leaders, and civil society organizations.

6. References:

[1] Face Recognition Module “Darpan” by Ayush Agnihotri.

[2] “Portrait Depth API: Turning a Single Image into a 3D Photo with TensorFlow.js”, by Ruofei Du, Yinda Zhang, Ahmed Sabie, Jason Mayes, Google on May 10, 2022

[3] “Body Segmentation with MediaPipe and TensorFlow.js”, by Ivan Grishchenko, Valentin Bazarevsky, Ahmed Sabie, Jason Mayes, Google, on January 31, 2022

[4] “Preparing Your Dataset for Machine Learning” on March, 2021

[5] “Study”, by Daniyal Malik, December 23, 2019 [Online] Available:
<https://www.digitalinformationworld.com/2019/12/new-password-study-finds-78-of-people-had-to-reset-a-password-they-forgot-in-past-90-days.html>

[6] “Content moderation using machine learning: a dual approach”, by Jen Person, Developer Advocate, August 19, 2022 [Online] Available: <https://blog.tensorflow.org/2022/08/content-moderation-using-machine-learning-a-dualapproach.html>

[7] “Portrait Depth API: Turning a Single Image into a 3D Photo with TensorFlow.js”, by Ruofei Du, Yinda Zhang, Ahmed Sabie, Jason Mayes, Google on May 10, 2022 [Online] Available:
<https://blog.tensorflow.org/2022/05/portrait-depth-api-turning-singleimage.html>

[8] “Body Segmentation with MediaPipe and TensorFlow.js”, by Ivan Grishchenko, Valentin Bazarevsky, Ahmed Sabie, Jason Mayes, Google, on January 31, 2022 [Online] Available:
<https://blog.tensorflow.org/2022/01/body-segmentation.html>

[9] “Preparing Your Dataset for Machine Learning” on March, 2021 [Online] Available:
<https://www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machinelearning-8-basic-techniques-that-make-your-data-better/>