

# **REAL TIME CREDIT CARD FRAUD DETECTION**

## **A PROJECT REPORT**

*Submitted by*

**VENKATESAN A                      812421104115**

**VIGNESH A                         812421104118**

**AJAY A                               812421104301**

**ELIYAS N                          812421104305**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**M.I.E.T. ENGINEERING COLLEGE**

**(Autonomous)**

**TIRUCHIRAPALLI-620 007**

**ANNA UNIVERSITY : CHENNAI 600 025**

**MAY 2025**

# **ANNA UNIVERSITY : CHENNAI 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**REAL TIME CREDIT CARD FRAUD DETECTION**” is the Bonafide work of “**VENKATESAN.A**”(812421104115), “**VIGNESH.A**”(812421104118), “**AJAY.A**”(812421104301), “**ELIYAS.N**”(812421104305) Who carried out the project work under the supervision.

### **SIGNATURE**

Dr. B. BHARATHI KANNAN M.Tech., Ph.D.,

### **HEAD OF THE DEPARTMENT**

Department of Computer Science

and Engineering

M.I.E.T. Engineering College.

Trichy-620007

### **SIGNATURE**

Mr. D. RAMACHANDRAN M.E.,

### **SUPERVISOR**

Assistant professor

Department of Computer Science

and Engineering.

M.I.E.T. Engineering College.

Trichy-620007

Submitted for the viva – voce examination held on .....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGMENT

First of all we thank God for this shower of blessing and his divine help which enables us to complete the project successfully.

We extend our sincere thanks to visionary and respected and highly esteemed **Alhaj. Janab.Er.A. MOHAMED YUNUS, B.E., M.Sc., (Engg.) Founder & Chairman** of M.I.E.T Institution, Trichy for offering the means of attaining our most cherished Goal Environment.

We extend our gratitude to Principal **Dr. A. NAVEEN SAIT, M.E.,Ph.D., M.I.E.T. Engineering College**, Trichy, for providing us permission to do the project work successfully.

We are grateful to express our profound thanks to the **Head of the department, Dr. B. BHARATHI KANNAN M.Tech., Ph.D.**, who has been the source of encouragement and moral strength throughout our study period.

It gives immense pleasure to extend my sincere and heartfelt gratitude to our **project Guide Mr. R. RAMACHANDRAN M.E.**, Assistant Professor for her valuable untiring and timely suggestions indispensable situation during the study period.

We are extremely thankful to our parents for enlightening us by providing Professional education and for their prayerful support that makes us to complete.

Also heartfelt thanks to our friends, Teaching and Non-teaching staff members who helped us to finish the project successfully.

## **ABSTRACT**

The most prevalent issue nowadays in the modern world is credit card fraud. This is due to the growth in internet transactions and e-commerce websites. When a credit card is stolen and used for unauthorized purposes, or when a fraudster uses the card's information for his own gain, credit card fraud happens. Because the credit card offers significant usage as a payment instrument, it is often used. As we all know, there are several opportunities for attackers or hackers to acquire sensitive data from online transactions. For both valid and invalid transactions, the information is processed and an acknowledgement is given to the bank. Facial detection and facial recognition technology employing the FaceSDK (FSDK) and Grassmann Algorithm will be used in a credit card transaction system. Attacks on several privacy concerns, such as credit cards, are the major issue that credit card users deal with. Typically, individuals experience this when their credit card is given to an unexpected party or misplaced. Therefore, we are developing a system that will lower the possibility of credit card fraud. The technology we're working on will compare the person's face in the photograph to the dataset for that user. A database will be kept for the purpose of authentication. If the photos line up, it signifies the user is real, and processing will be permitted; otherwise, the transaction will not be allowed.

## **TABLE OF CONTENT**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>iv</b>
	<b>LIST OF FIGURES</b>	<b>viii</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>x</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>2</b>
	2.1 A DUAL APPROACH FOR CREDIT CARD FRAUD DETECTION USING NEURAL NETWORK AND DATA MINING TECHNIQUES	2
	2.2 CREDIT CARD FRAUD DETECTION THROUGH MACHINE LEARNING ALGORITHM	3
	2.3 FRAUDULENT TRANSACTIONS DETECTION IN CREDIT CARD USING DATA MINING METHOD	4
	2.4 COMPARATIVE STUDY OF MACHINE LEARNING BASE CLASSIFICATION TECHNIQUE FOR CREDIT CARD FRAUD DETECTION	5
	2.5 RECOGNIZING CREDIT CARD FRAUD USING MACHINE LEARNING METHODS	7
<b>3</b>	<b>SYSTEM ANALYSIS</b>	<b>9</b>
	3.1 EXISTING SYSTEM	9
	3.1.1 DISADVANTAGE	9
	3.2 PROPOSED SYSTEM	10
	3.2.1 ADVANTAGE	10

<b>4</b>	<b>SYSTEM REQUIREMENT</b>	<b>12</b>
	4.1 SOFTWARE REQUIREMENT	12
	4.2 HARDWARE REQUIREMENT	12
<b>5</b>	<b>SYSTEM DESIGN</b>	<b>15</b>
	5.1 SYSTEM ARCHITECTURE	15
	5.2 DATA FLOW DIAGRAM	18
	5.3 USE CASE DIAGRAM	20
	5.4 CLASS DIAGRAM	22
	5.5 SEQUENCE DIAGRAM	22
	5.6 ACTIVITY DIAGRAM	24
<b>6</b>	<b>MODULE DESCRIPTION</b>	<b>25</b>
	6.1 LIST OF MODULES	25
	6.1.1 ADMIN LOGIN MODULE	25
	6.1.2 ADD EMPLOYEE / PRODUCT MODULE	25
	6.1.3 VIEW BOOKING DETAILS MODULE	26
	6.1.4 VIEW USER DETAILS MODULE	26
	6.1.5 USER REGISTER MODULE	26
	6.1.6 LOGIN MODULE	26
	6.1.7 PRODUCT PURCHASE MODULE	26
	6.1.8 FACE RECOGNIZE MODULE	26
	6.1.9 MAKE PAYMENT MODULE	26
<b>7</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>27</b>
	7.1 CONCLUSION	27
	7.2 FUTURE ENHACEMENT	27
	7.2.1 INFORM THE BANK IMMEDIATELY	27
	7.2.2 ALTERNATIVE VERIFICATION	28

	METHODS	
	7.2.3 UPDATE FACE DATA(IF APPLICABLE)	28
<b>8</b>	<b>APPENDIX I &amp; II</b>	<b>29</b>
	8.1 SOURCE CODE	29
	8.2 SCREENSHOTS	55
<b>9</b>	<b>REFERENCES</b>	<b>66</b>

## **LIST OF FIGURES**

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
5.1	SYSTEM ARCHITECTURE	17
5.2	DATA FLOW DIAGRAM(LEVEL 0)	18
5.3	DATA FLOW DIAGRAM(LEVEL 1)	19
5.4	DATA FLOW DIAGRAM(LEVEL 2)	20
5.5	USE CASE DIAGRAM	21
5.6	CLASS DIAGRAM	22
5.7	SEQUENCE DIAGRAM	23
5.8	ACTIVITY DIAGRAM	24
A.2.1	NEW USER REGISTRATION	55
A.2.2	BIOMETRIC BY FACE VERIFICATION	55
A.2.3	USER LOGIN	56
A.2.4	OTP VERIFICATION	56
A.2.5	HOME PAGE	57
A.2.6	PRODUCT PURCHASE	57
A.2.7	PERSONAL INFORMATION	58
A.2.8	PRODUCT INFORMATION	58
A.2.9	PRODUCT CONFIRMATION	59
A.2.10	CART INFORMATION	59
A.2.11	FACE RECOGNITION BY LIVENESS	60
A.2.12	PRODUCT & PAYMENT INFORMATION	60
A.2.13	ADMIN LOGIN	61
A.2.14	USER INFORMATION	61



A.2.15	PRODUCT INFORMATION	62
A.2.16	APPROVED FOR RENT INFORMATION	62
A.2.17	PAYMENT INFORMATION	63
A.2.18	NEW SHOP KEEPER REGISTRATION	63
A.2.19	SHOP KEEPER LOGIN PAGE	64
A.2.20	PERSONAL INFORMATION	64
A.2.21	NEW PRODUCT REGISTRATION	65
A.2.22	PRODUCT INFORMATION	65

## **LIST OF ABBREVIATIONS**

<b>FR</b>	Facial Recognition
<b>OTP</b>	One-Time Password
<b>FSDK</b>	Face Software Development Kit
<b>LD</b>	Liveness Detection
<b>2FA</b>	Two-Factor Authentication
<b>ML</b>	Machine Learning
<b>UML</b>	Unified Modeling Language
<b>OpenCV</b>	Open Source Computer Vision Library.
<b>DFD</b>	Data Flow Diagram
<b>IDE</b>	Integrated Development Environment
<b>RDBMS</b>	Relational Database Management System
<b>GDA</b>	Grassmann Discriminant Analysis

# **CHAPTER 1**

## **INTRODUCTION**

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. For example, the objective of project is to develop a general purpose e-commerce store where any product (such as books, CDs, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet. An online store is a virtual store on the Internet where customers can browse the catalogue and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. Proposed System the credit card transaction is using face recognition technology. If the face detection process is completely finish then only move on to the next process. An e- mail notification is sent to the customer as soon as the order is placed. Seller selling products on the web often ask or take reviews from customers about the products that they have purchased This creates difficulty for the potential customer to read them and to make a decision whether to buy or not the product.. And also additional difficulties are faced by the manufacturer because many other merchant sites may sell the same product at good ratings and the manufacturer normally produces many kinds of products.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 TITLE: A DUAL APPROACH FOR CREDIT CARD FRAUD DETECTION USING NEURAL NETWORK AND DATA MINING TECHNIQUES**

**Author:** SAHU, AANCHAL (2020)

Build new models to detect fraudulent credit card transactions using five classifiers to find out the best fit classifier for the situation. The dataset contains the details of some European credit card holders recorded in September 2013. The dataset comprises the transaction information for two days, which has 492 fraud transactions out of 284,807 transactions. For security purposes, the features of the dataset are not revealed. Instead, the PCA values of the features are given. A PCA is a technique to get a low dimensional structure out of a potential high dimensional dataset. It includes the extraction of  $q$  eigenvectors for  $q$  input distribution [10]. It is one of the most used algorithm for dimensionality reduction. The basis vectors are known as principal components. The dataset contains a total of 31 columns of which, 28 are PCA components named as V1, V2.V28. Moreover, the time and amount of the money transaction has also been provided. The target variable classifies a transaction as 0 for valid transaction and 1 for fraudulent transaction. Here apply classifiers of logistic regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF) and artificial neural network (ANN) on two different approaches used on the same data. In both the approaches, the main goal was to curb the problem of data imbalance (since number of fraudulent cases are scarce in comparison to normal behaviour). In the first approach we resampled the minority class to a higher number close to the number of samples of normal class, while, in the second approach we use cost-based methods such as applying weights on classes so that the minority class has a higher impact on the loss (error)

calculation for the models. After thorough experimentation, we notice that RF outperforms all other classifiers on an average on both the approaches.

**ADVANTAGE:**

- Using resampling and cost-based methods helps the models to better identify rare fraudulent transactions.
- Improving overall detection accuracy.

**DISADVANTAGE:**

- The complex challenges are hard to resolve.

## **2.2 TITLE: CREDIT CARD FRAUD DETECTION THROUGH MACHINE LEARNING ALGORITHM**

**Author:** PANDA, AGYAN (2021)

Implement credit card fraud detection methods based on data mining. Classic data mining algorithms aren't directly applicable to our topic because it's handled as a classification challenge. As a result, a different technique is employed, which involves the employment of general-purpose meta heuristics like genetic algorithms. The purpose of this study is to create a credit card fraud detection system based on genetic algorithms. Genetic algorithms are a form of evolutionary algorithm that tries to continuously improve solutions. When a card is duplicated, stolen, or lost by fraudsters, it is usually utilized until the available limit is exhausted. As a result, rather than focusing on the quantity of correctly classified transactions, a strategy that reduces the overall allowed limit on fraud-prone cards takes precedence. Its goal is to reduce false alerts by utilizing a genetic algorithm to optimize a set of interval-valued parameter. This concept is difficult to put into practice in practice since it necessitates the collaboration of banks, which are unwilling to exchange information owing to market

competition, as well as for legal concerns and the protection of their users' data. As a consequence, we searched up some reference publications that used comparable methods and gathered data. As stated in one of these reference papers: Credit card fraud is unquestionably a kind of criminal deception. This article evaluated current results in this subject and outlined the most prevalent types of fraud, as well as how to identify them. This study also goes into great depth on how machine learning may be used to improve fraud detection outcomes. Pseudo code, explanation its implementation and experimentation results.

#### **ADVANTAGE:**

- GA helps to optimize parameters and find the best rules for fraud detection.
- Which can improve accuracy and adaptability over time.

#### **DISADVANTAGE:**

- The complex challenges are hard to resolve.

### **2.3 TITLE: FRAUDULENT TRANSACTIONS DETECTION IN CREDIT CARD BY USING DATA MINING METHODS**

**Author:** AZIZ, AMIR, AND HAMID GHOU (2021)

This work provides the comparative study of techniques to uncover and detect the ways of fraud by following machine learning (ML) methods like Random Forest (RF), Deep Learning (DL), Support Vector Machine (SVM), Hybrid Methods (HM), and Decision Tree (DT). All these techniques are used to discover common usage patterns of consumers as following their past activities. After reviewing ML methods, it has been revealed tremendous discrepancies amongst different studies for future works. As discussed in this comparative study, some issues have been addressed whenever a transaction

occurs while others are remained to be located that shows the direction of the future for highlighting and focusing of attention in the area of CCFD. After exploring and examining the limitations some extensions can be useful to improve the accuracy and other measures to detects frauds in credit cards. Some effective methods of data pre-processing may be useful, such as sampling methods, clustering algorithms, and some advanced methods of selection of features. For the Selection of Features, the majority voting method could be used, the Genetic Algorithm could be used to decrease the dataset, the optimized parameter selection kernel function can be used, and DL methods could be used for pre-processing. One of these methods, or a combination of them, can be used to detect fraud. Most researchers face some challenges, such as unavailability of real datasets, unbalanced datasets, and size of datasets, which is a difficult subject of research to detect credit card fraud.

#### **ADVANTAGE:**

- By analyzing various models (RF, DL, SVM, HM, DT), the study provides insights into strengths and weaknesses.
- Aiding researchers in selecting the most suitable model for their needs.

#### **DISADVANTAGE:**

- SVM model is challenging to comprehend.

### **2.4 TITLE: COMPARATIVE STUDY OF MACHINE LEARNING BASED CLASSIFICATION TECHNIQUES FOR CREDIT CARD FRAUD DETECTION**

**Author:** SHAH, ANKIT, AND AKASH MEHTA (2021)

Implement six widely used machine learning techniques for credit card fraud detection. For each machine learning technique, a confusion matrix is prepared for performance analysis of the algorithm. Their efficacy is analyzed based on the

parameters such as accuracy, precision, recall, specificity, misclassification, and F1 score. Results of fraud detection techniques also depend on a type of dataset. Some techniques give high accuracy but training them is very expensive. With small data sets, some techniques give excellent results, but they do not apply to large datasets. With sampled and pre-processed data, some techniques give better results whereas some techniques give better accuracies with raw unsampled data. It is also important to note that the outlier class of modelling can be senseless and unproductive in solving the problem of anomaly detection. There is a need to focus on the structure of the normal data and its distribution. Credit card fraud detection system should be capable of detecting fraud in the transit process and to identify fraud precisely and wrong classifications should have to be minimum. There is a need for a technology that should be able to detect fraudulent transactions when it is occurring. The best result cannot be achieved by applying one machine learning technique. Therefore, to achieve better performance for credit card fraud detection the integration of multiple techniques may be used. In the future, researchers can compute the computational complexity and execution time of various algorithms to suggest the best possible solution.

**ADVANTAGE:**

- The study uses a comprehensive set of evaluation metrics—accuracy, precision, recall, specificity.
- F1 score, and misclassification rate—providing a well-rounded performance analysis of each algorithm.

**DISADVANTAGE:**

- K-NN requires a lot of computing.



## **2.5 TITLE: RECOGNIZING CREDIT CARD FRAUD USING MACHINE LEARNING METHODS**

**Author:** MUTTIPATI, APPALA SRINUVASU (2021)

This proposed work going to address the problem of an imbalanced dataset. SMOTE sampling technique is used to convert the imbalanced dataset to a balanced binary dataset. The credit card dataset which we have chosen for reference dataset consists of an error of 0.172 percent. It gives a meaning that, the referenced dataset contains 0.172 percentage of no genuine transactions. This infers that, the dataset is uneven and is prejudiced towards genuine transaction. Because of the bias the network is unable to identify and could give a correct prediction of the error. This problem can be solved by using 2 techniques, i) under-sampling ii) over-sampling techniques to condense the partiality for accurate results. Under-sampling technique, balances the dataset by basing on the non-bias class i.e. fraudulent Transactions. By adjusting, total of genuine transactions on equality with fraudulent transactions by removing the excess genuine values from the data. For example, suppose there is an 100 observations, then 7 fraudulent values give a 7% error. Similar to that we compute the total of genuine transactions for 492 fraudulent ones, by removing the excess. This produces data with 3% error, it become because easier to detection process. By utilizing this method it results to loss of information. Over-sampling is another technique that is utilized for imbalanced to balanced data. Here, the occurring of bias is due to a replica of information in terms of the recurring rows but not for the loss of information. We need to eradicate the bias. For sample, here recurring non-genuine transitions are added from 492. A total of observations should give an error of 3%. In this context

instead of removing, adding a more number of observations. Hence, by utilizing this technique, we can achieve a high-accuracy model.

**ADVANTAGE:**

- The use of SMOTE (Synthetic Minority Over-sampling Technique), along with under-sampling and over-sampling.
- It helps create a balanced dataset.
- Improving model sensitivity to fraudulent transactions.

**DISADVANTAGE:**

- Susceptible to extreme cases.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

Most current payment systems mainly depend on OTP-based authentication, which is susceptible to cyber threats like SIM swapping and phishing. These attacks can easily bypass OTPs, compromising user accounts. Unfortunately, many systems do not integrate biometric verification, such as fingerprint or facial recognition, to verify user identity. This lack of robust security increases the risk of unauthorized access. Moreover, fraud detection usually happens after the transaction is completed. This reactive approach fails to stop fraud in real time. E-commerce platforms, in particular, struggle to provide adequate fraud prevention. Users and businesses are left vulnerable to financial losses. Strengthening authentication methods is crucial to improve payment security.

##### **3.1.1 DISADVANTAGES:**

- **Vulnerability to SIM Swapping:** Attackers can hijack a user's phone number and receive OTPs, allowing unauthorized access to accounts.
- **Susceptibility to Phishing:** Users may unknowingly share OTPs with attackers through fake websites or messages.
- **Lack of Biometric Verification:** Without biometric checks, systems cannot fully confirm the user's identity, increasing fraud risk.
- **Delayed Fraud Detection:** Fraudulent transactions are often identified only after they occur, leading to financial loss before action can be taken.

## 3.2 PROPOSED SYSTEM

To enhance the security of online credit card transactions by integrating OTP verification and facial recognition. Initially, users must log in using a one-time password (OTP) sent to their registered mobile number, adding a layer of protection against phishing and SIM swap attacks. Once authenticated, users can browse and select products for purchase. During checkout, the system requires facial verification to ensure that the transaction is being made by the legitimate user. OpenCV is used to detect the face, and FaceSDK (FSDK) extracts and analyzes facial features. These features are then compared with the stored user face data using the Grassmann algorithm. To prevent spoofing, liveness detection verifies that the face is from a live person, not a photo or video. Only when both OTP and facial verification succeed, the system permits the transaction. This dual-authentication method significantly reduces the chances of credit card fraud. As a result, it ensures secure, real-time processing while maintaining user trust in e-commerce platforms..

### 3.2.1 ADVANTAGES:

- **Two-Factor Authentication (2FA):**  
Combines OTP and facial recognition to ensure only authorized users can access and complete transactions.
- **Enhanced Security:**  
Reduces risks from phishing, SIM swap attacks, and stolen card credentials through layered verification.
- **Real-Time Fraud Detection:**  
Detects and blocks unauthorized access instantly during login or payment.

- **User-Friendly Experience:**  
Seamless integration of OTP and face authentication provides security without complex user steps.
- **Liveness Detection:**  
Prevents face spoofing using photos or videos, ensuring only live users are verified.
- **Automated Face Verification:**  
FaceSDK accurately identifies users by comparing live and registered facial features in real time.
- **Compliance with Financial Security Norms:**  
Supports regulations like RBI's strong customer authentication (SCA) and global 2FA best practices.
- **Builds Trust in E-Commerce Platforms:**  
Customers gain confidence knowing their transactions are protected with advanced biometric technology.

## **CHAPTER 4**

### **SYSTEM REQUIREMENTS**

#### **4.1 SOFTWARE REQUIREMENTS**

- Operating system : Windows OS
- Back End : Python
- Database : MySQL SERVER
- IDE : PyCharm

#### **4.2 HARDWARE REQUIREMENTS**

- Processor : Intel Processor
- RAM : 2GB
- Hard disk : 160 GB
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

#### **DEFINITIONS:**

##### **PYCHARM:**

PyCharm is an integrated development environment (IDE) developed by JetBrains specifically for programming in Python. It provides a wide range of features to help developers write clean, efficient, and maintainable code.

##### **FLASK:**

Flask is a micro web framework written in Python that allows developers to build web applications easily. It is lightweight, easy to use, and provides flexibility by not enforcing a specific project structure or tools.

## **OPENCV:**

OpenCV (Open Source Computer Vision Library) is an open-source library in C++ with Python bindings, used for computer vision, image processing, and machine learning tasks. It helps in tasks like face detection, object recognition, and video analysis.

## **HTML & CSS:**

- HTML (Hypertext Markup Language) is the standard language used to create and structure content on the web, such as text, images, and links.
- CSS (Cascading Style Sheets) is used to style and design the appearance of HTML elements, including layout, colors, fonts, and spacing.
- Together, HTML builds the structure, and CSS makes it visually appealing.

## **JAVASCRIPT:**

JavaScript is a scripting language used to create dynamic and interactive content on websites. It enables features like animations, form validations, real-time updates, and user interactions, working alongside HTML and CSS to enhance web pages.

## **MySQL:**

MySQL is an open-source relational database management system (RDBMS) used to store, manage, and retrieve data efficiently. It uses Structured Query Language (SQL) for accessing and managing databases.

**GRASSMANN:**

The Grassmann algorithm generally refers to methods based on Grassmann algebra or the Grassmann manifold, used to represent and compute with subspaces of vector spaces. These algorithms are applied in areas like signal processing, computer vision, and machine learning to compare and analyze data lying in different subspaces.

**FSDK:**

FSDK stands for Face SDK (Software Development Kit), which is typically a library or set of tools used for face detection and recognition in various applications. It helps developers integrate facial recognition capabilities into their software by providing features like detecting faces in images, comparing faces, and identifying individuals.

**LIVENESS:**

Liveness refers to the ability to distinguish between a real, live person and a static representation (like a photo, video, or mask) in biometric systems, especially in face recognition and fingerprint technologies. Liveness detection ensures that the biometric data being captured is from an actual person present at the time of authentication, preventing spoofing or fraudulent attempts to bypass security systems. It often involves analyzing subtle movements, texture, or patterns that can only be detected from a live subject.



## **CHAPTER 5**

### **SYSTEM DESIGN**

#### **5.1 SYSTEM ARCHITECTURE:**

##### **Architectural Design**

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behaviour) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

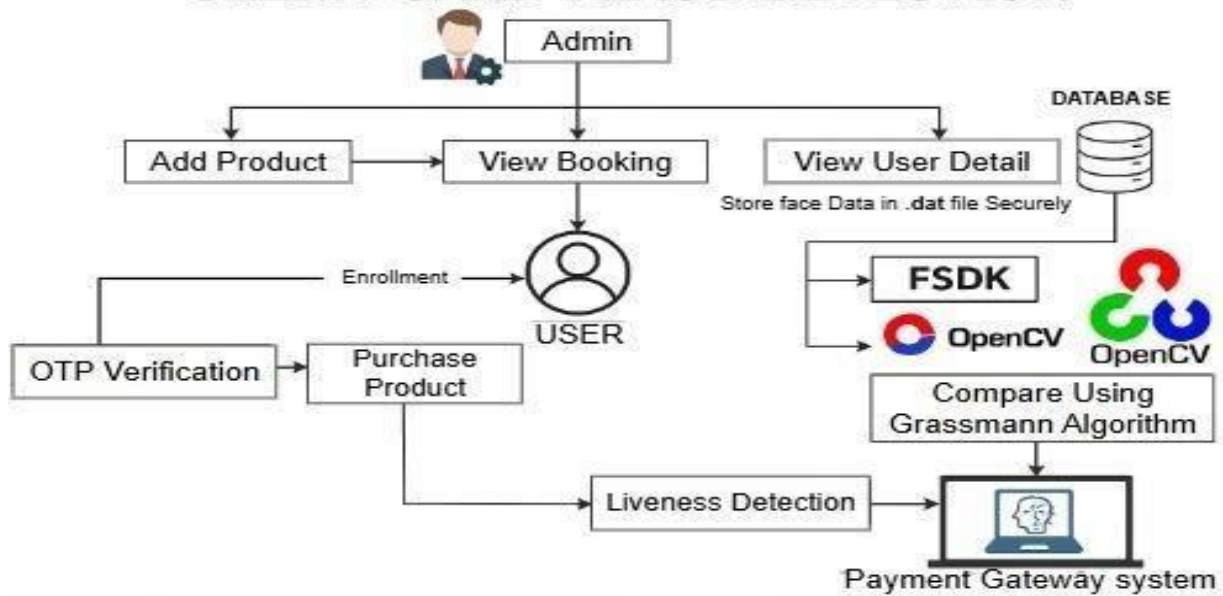
**Various organizations define systems architecture in different ways, including:**

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of

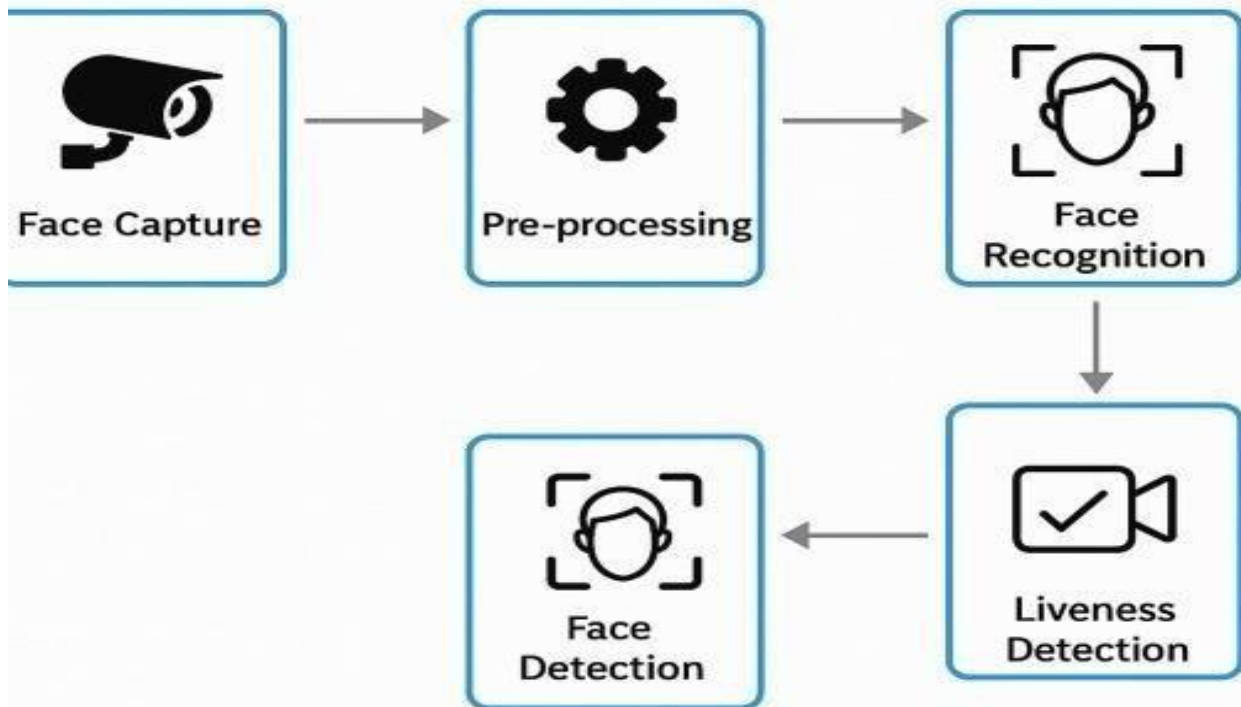
long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software

- The composite of the design architectures for products and their life-cycle processes.

## CREDIT CARD FRAUD DETECTION



## WORKING PROCESS OF FSDK & OPENCV



**Figure 5.1 System Architecture**

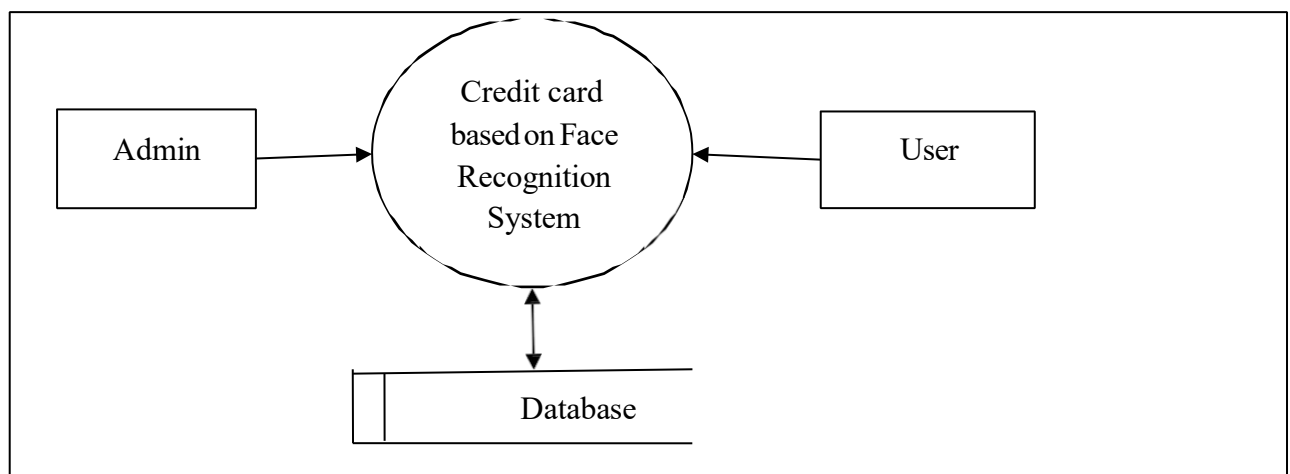
## 5.2 DATA FLOW DIAGRAM:

### Data Flow Diagram:

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

### LEVEL 0:

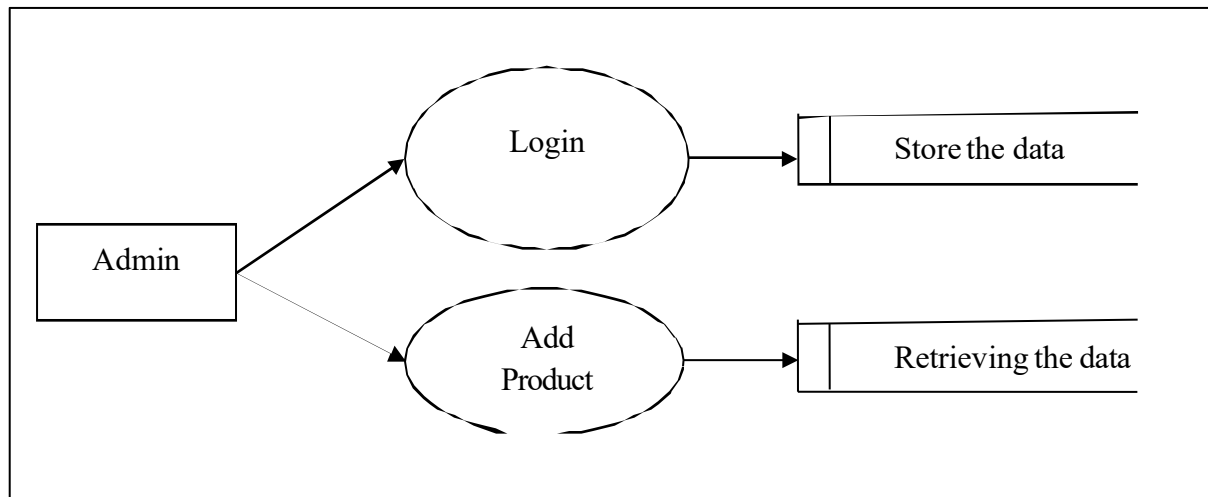
The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



**Figure 5.2 DFD Level 0**

## LEVEL 1:

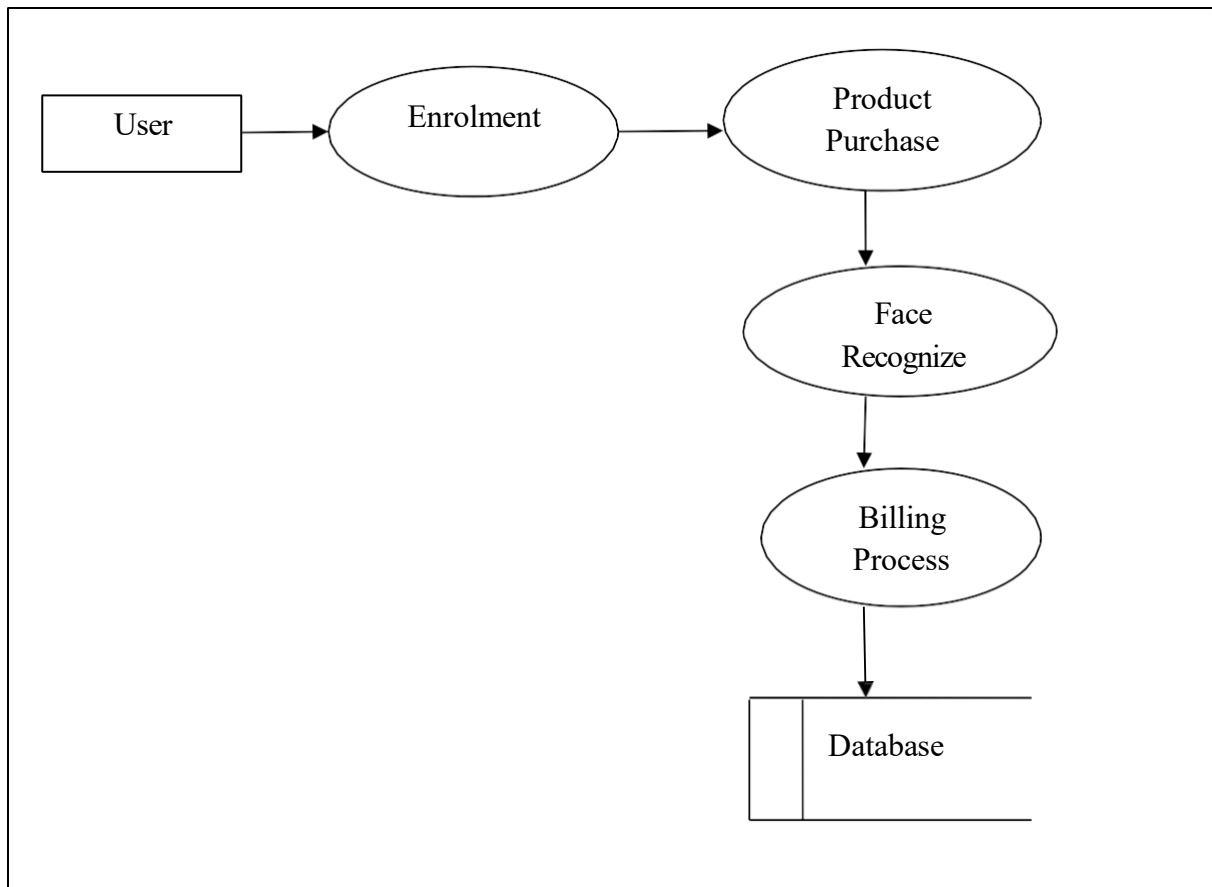
The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.



**Figure 5.3 DFD Level 1**

## DFD LEVEL 2:

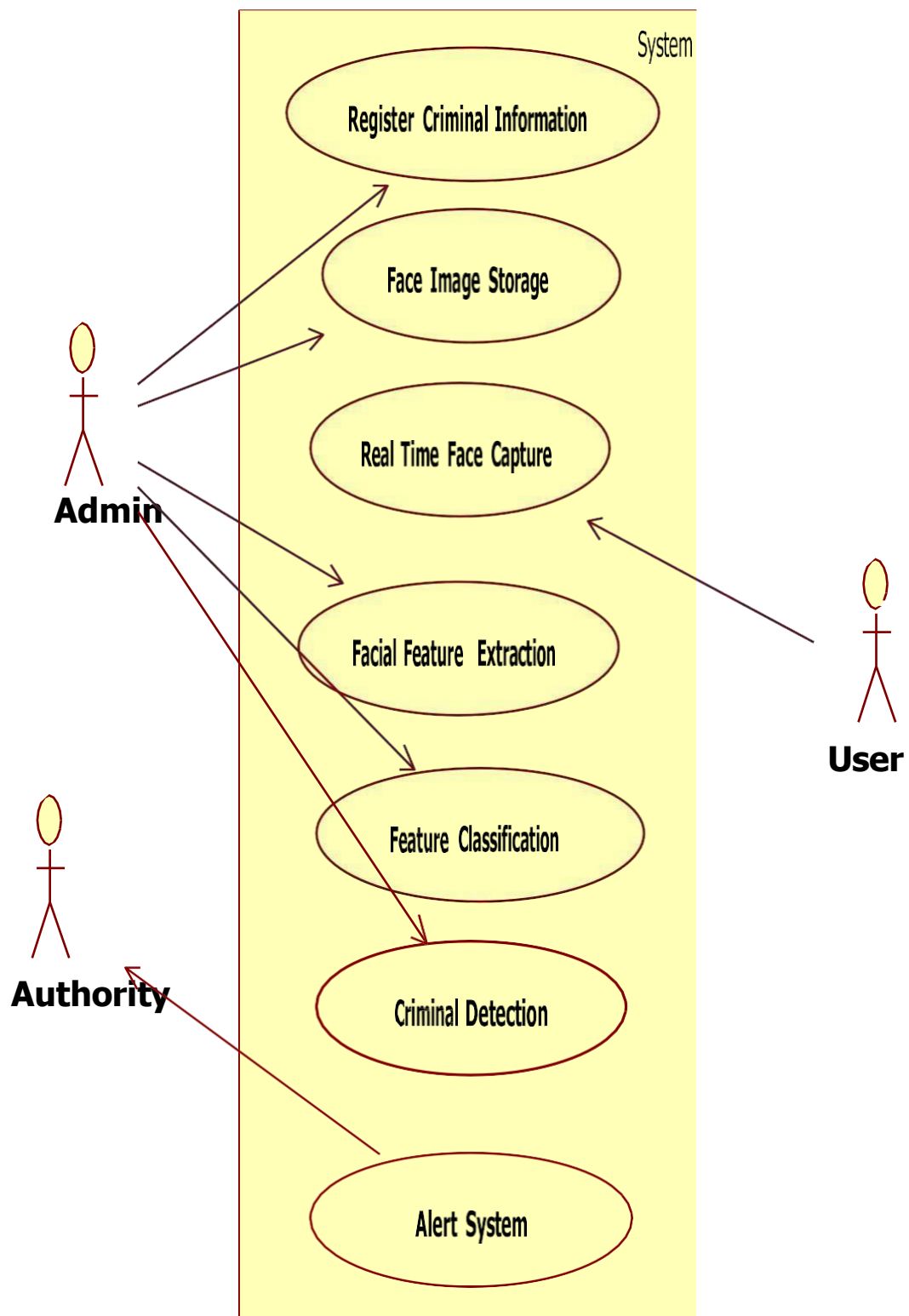
A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data within the defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modelling and one of the oldest.



**Figure 5.4 DFD Level 2**

### **5.3 USE CASE DIAGRAM:**

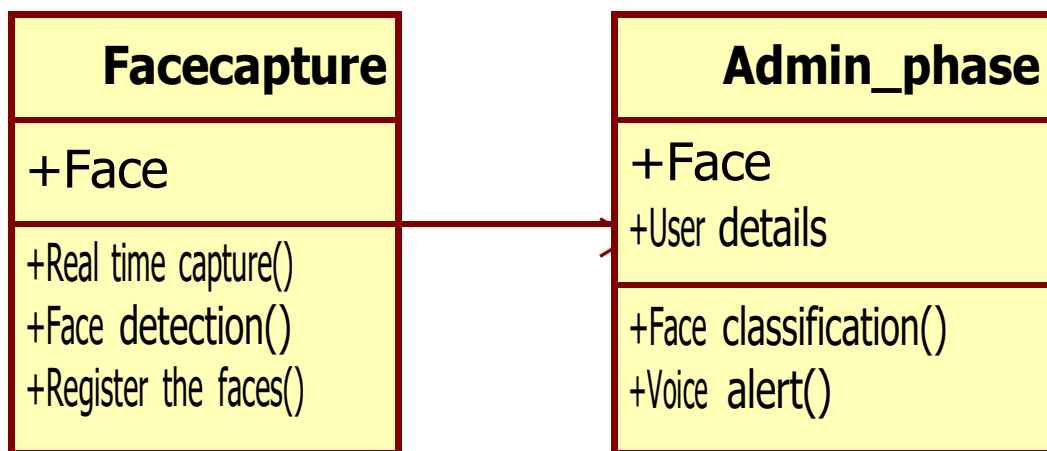
In its most basic form, a use case diagram is a depiction of a user's interaction with the system that illustrates the connection between the user and the many use cases that the user is involved in. A "system" in this sense refers to something that is being created or run, like a website. The "actors" are individuals or groups functioning inside the system in designated roles.



**Figure 5.5 Use Case Diagram**

## 5.4 CLASS DIAGRAM:

A class diagram, as defined by the Unified Modelling Language (UML), is a kind of static structural diagram that illustrates a system's classes, properties, functions, and interactions between objects. The fundamental component of object-oriented modelling is the class diagram. It is utilized for both technical modelling—which converts the models into computer code—and general conceptual modelling of the applications systematic.

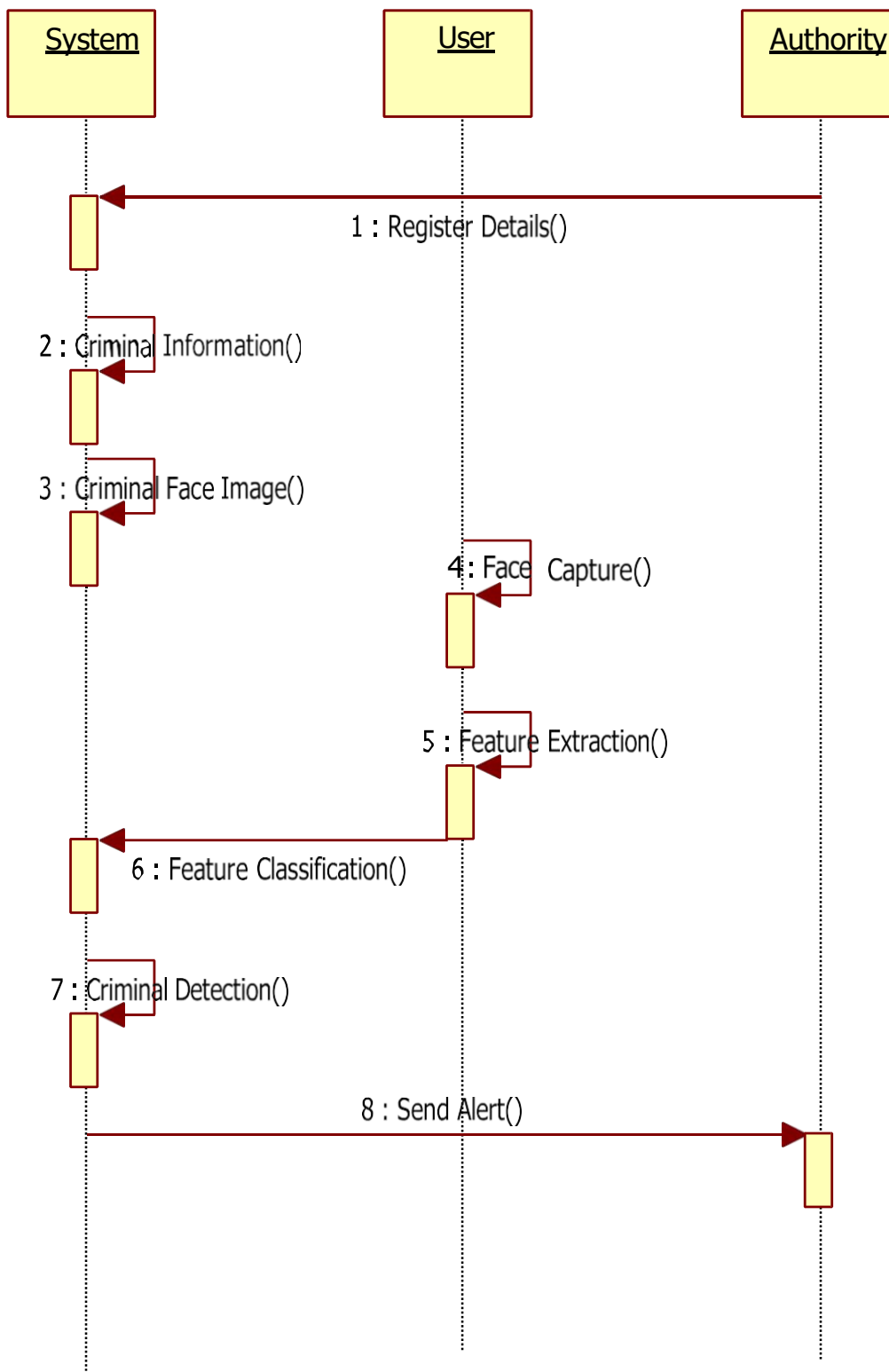


**Figure 5.6 Class Diagram**

## 5.5 SEQUENCE DIAGRAM:

An object's interactions are arranged chronologically in a sequence diagram. It shows the classes and objects that are a part of the scenario as well as the messages that are passed between the objects in order for the scenario to work. Sequence diagrams are commonly linked to the realizations of use cases in the Logical View of the system that is being developed. Event diagrams or event scenarios are other names for sequence diagrams.

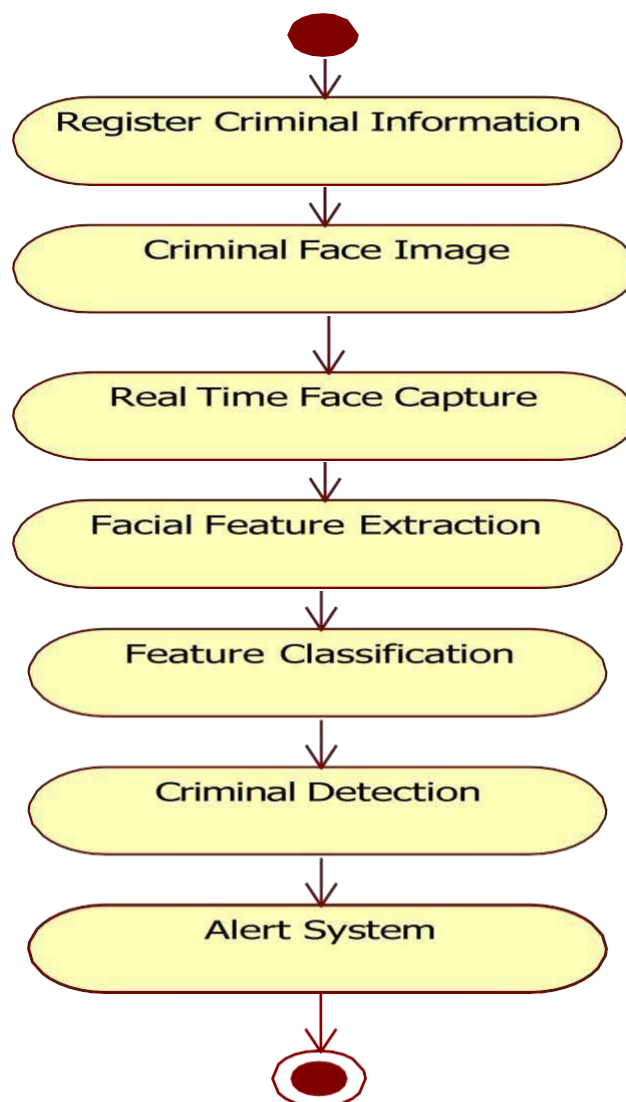




**Figure 5.7 Sequence Diagram**

## 5.6 ACTIVITY DIAGRAM

The activity diagram shows a unique kind of state diagram in which the majority of states are action states and the majority of transitions are brought about by the fulfillment of actions in the source states. One may refer to the action as a system operation. As a result, the control flow is transferred across operations. This flow may occur concurrently, forked, or sequentially. Activity diagrams use a variety of features to address various forms of flow control.



**Figure 5.8 Activity Diagram**

## **CHAPTER 6**

### **MODULE DESCRIPTION**

#### **6.1 LIST OF MODULES**

##### **Admin**

- Login Module
- Add Employee/ Product Module
- View booking Details Module
- View User Details Module

##### **User**

- Register Module
- Login Module
- Product Purchase Module
- Face Recognize Module
- Make Payment Module

#### **6.1.1 ADMIN LOGIN MODULE**

In this module, the admin can login in the system using his/her username and password.

#### **6.1.2 ADD EMPLOYEE/ PRODUCT MODULE**

In this module, the admin can add the employee information like employee name, id, phone number, mail id, location etc. After the login process the employee can add the product details like product id, name, type, amount, quantity and so on.

### **6.1.3 VIEW BOOKING DETAILS MODULE**

In this module, the admin/employee can view the user booking details. The booking details contain booking id, product details, user details etc.

### **6.1.4 VIEW USER DETAILS MODULE**

In this module, the admin can view the user information's like user name, email, gender, mobile number, address, and etc.

### **6.1.5 USER REGISTER MODULE**

There is registration form available where new user can create their account by providing required information to the system. The registration form details are like user name, email, gender, mobile number, address, and etc. These details are stored in the database. And then can getting to the username and password in the system.

### **6.1.6 LOGIN MODULE**

In this module, user can login in the system using his/her username and password.

### **6.1.7 PRODUCT PURCHASE MODULE**

In this module, the user can view the product details like product name, type, amount, description etc. After viewing all products, the user can buy product using this module. The user buying details are sent to the employee/admin.

### **6.1.8 FACE RECOGNIZE MODULE**

After successfully entered the card details the system camera can capture the user face image to match the particular card holder account database. If the user face image is matched with database, the user payment is transferred. Otherwise the user payment is not transferred.

### **6.1.9 MAKE PAYMENT MODULE**

In this module used to make payment. This module contains user's card details like name, card no, amount etc.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **7.1 CONCLUSION**

This project entitled as “Credit Card Transaction Based on Face Recognition Technology” has been developed to satisfy all the proposed requirements. The process of recording details about online shopping is more simple and easy. The system reduces the possibility of errors to a great extent and maintains the data in an efficient manner. User friendliness is the unique feature of this system. The system generates the reports as and when required. The system is highly interactive and flexible for further enhancement. The coding is done in a simplified and easy to understandable manner so that other team trying to enhance the project can do so without facing much difficulty. The documentation will also assist in the process as it has also been carried out in a simplified and concise way

#### **7.2 FUTURE ENHANCEMENT**

If someone’s face is damaged (due to injury, burns, surgery, etc.) and they need to apply at a bank where face detection or facial recognition is required (e.g., for KYC, authentication, or account access), there are a few key points to consider:

##### **7.2.1 Inform the Bank Immediately**

- Notify the bank staff or branch manager about the facial injury.
- Provide medical documentation if available, to explain the change in facial appearance.

### **7.2.2 Alternative Verification Methods**

**Banks usually offer alternative methods of identification when facial recognition fails:**

- Manual KYC verification: Using Aadhaar, PAN, passport, or other government-issued IDs.
- Biometric authentication: Such as fingerprint or iris scan (if supported).
- OTP-based authentication: Sent to registered mobile/email.
- Physical verification: Visiting the branch in person for identity confirmation.

### **7.2.3 Update Face Data (if applicable)**

- Once healed or as recommended by your doctor, you can request to update your facial data at the bank.
- Banks using biometric systems may allow you to re-enroll or re-capture your face data for future use.

## APPENDIX I

### 8.1 SOURCE CODE

```
from flask import Flask, render_template, request, session, flash
from flask import Flask, render_template, flash, request, session
from flask import render_template, redirect, url_for, request
import sys, fsdk, math, ctypes, time
import mysql.connector
import hmac
import hashlib
import binascii
import random
import datetime

app = Flask(__name__)
app.config['SECRET_KEY'] = 'aaa'

@app.route('/')
def home():

    conn = mysql.connector.connect(user='root', password='', host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM protb ")

    data = cur.fetchall()

    return render_template('index.html', data=data)

@app.route('/AdminLogin')
def AdminLogin():
```

```

return render_template('AdminLogin.html')

@app.route('/NewUser')
def NewUser():
    return render_template('NewUser.html')

@app.route('/UserLogin')
def UserLogin():
    return render_template('UserLogin.html') @app.route('/NewShopKeeper')
def NewShopKeeper():
    return render_template('NewShopKeeper.html')

@app.route('/ShopKeeperLogin')
def ShopKeeperLogin():
    return render_template('ShopKeeperLogin.html')

@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None

    if request.method == 'POST':
        if request.form['uname'] == 'admin' and request.form['password'] == 'admin':

            conn = mysql.connector.connect(user='root', password="", host='localhost',
            database='25onlinemobiledb')

            cur = conn.cursor()

            cur.execute("SELECT * FROM regtb ")

            data = cur.fetchall()

            flash("you are successfully Login")

            return render_template('AdminHome.html', data=data)

```



```

else:

flash("UserName or Password Incorrect!")

return render_template('AdminLogin.html')

@app.route('/NewProduct')

def NewProduct():

return render_template('NewProduct.html')

@app.route("/AdminHome")

def AdminHome():

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM regtb ")

data = cur.fetchall()

return render_template('AdminHome.html', data=data)

@app.route("/newproduct", methods=['GET', 'POST'])

def newproduct():

if request.method == 'POST':

pname = request.form['pname']

ptype = request.form['ptype']

price = request.form['price']

offer = request.form['offer']

info = request.form['info']

qty = request.form['qty']

import random

file = request.files['file']

```

```

fnew = random.randint(1111, 9999)

savename = str(fnew) + ".png"

file.save("static/upload/" + savename)

print(price)

print(offer)

offeramount = float(price) * (float(offer) / 100)

print(offeramount)

total = float(price) - float(offeramount)

print(total)

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor=conn.cursor()

cursor.execute(

"insert into protb values(", "" + pname + ", "" + ptype + ", "" + str(

price) + ", "" + str(offer) + ", "" + str(total) + ", "" + info + ", "" + savename + ", "" +

str(

qty) + ")")

conn.commit()

conn.close()

flash('Product Info Save Successfully!')

return render_template('NewProduct.html')

@app.route("/ProductInfo")

def ProductInfo():

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

```

```

cur = conn.cursor()

cur.execute("SELECT * FROM protb ")

data = cur.fetchall()

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM protb where Qty <= 4 ")

data1 = cur.fetchall()

msg = ""

iii = 0

for item1 in data1:

pname = "ProductName : " + item1[1] + ""

qty = "Quantity : " + str(item1[8]) + "\n"

msg += pname + qty

iii = 1

#print(msg)

if iii == 1:

sendmsg("sangeeth5535@gmail.com", msg)

return render_template('ProductInfo.html', data=data)

def sendmsg(Mailid, message):

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.base import MIMEBase

from email import encoders

```

```

fromaddr = "projectmailm@gmail.com"

toaddr = Mailid

# instance of MIMEMultipart
msg = MIMEMultipart()

# storing the senders email address
msg['From'] = fromaddr

# storing the receivers email address
msg['To'] = toaddr

# storing the subject
msg['Subject'] = "Alert"

# string to store the body of the mail
body = message

# attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login(fromaddr, "qmgn xecl bkqv musr")

# Converts the Multipart msg into a string
text = msg.as_string()

#      sending      the      mail
s.sendmail(fromaddr, toaddr, text)

```

```

# terminating the session

s.quit()

@app.route("/SProductInfo")

def SProductInfo():

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM protb ")

    data = cur.fetchall()

    return render_template('SProductInfo.html', data=data)

@app.route("/SalesInfo")

def SalesInfo():

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM carttb where Status='1'")

    data1 = cur.fetchall()

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM booktb ")

    data2 = cur.fetchall()

    return render_template('SalesInfo.html', data1=data1, data2=data2)

@app.route("/Remove")

def Remove():

```

```

id = request.args.get('id')

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor= conn.cursor()

cursor.execute(

"delete from protb where id='" + id + "'")

conn.commit()

conn.close()

flash('Product info Remove Successfully!')

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM protb ")

data = cur.fetchall()

return render_template('ProductInfo.html', data=data)

@app.route("/hsearch")

def hsearch():

id = request.args.get('id')

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM protb where ProductType='" + id + "' ")

data = cur.fetchall()

return render_template('index.html', data=data)

@app.route("/newuser", methods=['GET', 'POST'])

```

```

def newuser():
    if request.method == 'POST':
        name = request.form['name']
        mobile = request.form['mobile']
        email = request.form['email']
        address = request.form['address']
        username = request.form['uname']
        password = request.form['password']

        conn = mysql.connector.connect(user='root', password="", host='localhost',
        database='25onlinemobiledb')

        cursor = conn.cursor()

        cursor.execute(
            "insert into regtb values('" + name + "','" + mobile + "','" + email + "','" + address
            + "','" + username + "','" + password + "')"

        conn.commit()

        conn.close()

    import LiveRecognition as liv

    # liv.att()

    del sys.modules["LiveRecognition"]

    flash("Record Saved!")

    return render_template('UserLogin.html')

@app.route("/newkeeper", methods=['GET', 'POST'])
def newkeeper():
    if request.method == 'POST':
        name = request.form['name']

```

```

mobile = request.form['mobile']

email = request.form['email']

address = request.form['address']

username = request.form['uname']

password = request.form['password']

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor= conn.cursor()

cursor.execute(

"insert into keepertb values("," + name + "," + mobile + "," + email + "," +
address + "," + username + "," + password + ")")

conn.commit()

conn.close()

flash("Record Saved!")

return render_template('NewShopKeeper.html')

@app.route("/keeperlogin", methods=['GET', 'POST'])

def keeperlogin():

if request.method == 'POST':

username = request.form['uname']

password = request.form['password']

session['uname'] = request.form['uname']

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

cursor.execute("SELECT * from keepertb where username=" + username + "
and password=" + password + "")

```



```

data = cursor.fetchone()

if data is None:

    flash('Username or Password is wrong')

    return render_template('NewShopKeeper.html', data=data)

else:

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM keepertb where username='" + username + "' and
    password='" + password + "'")

    data = cur.fetchall()

    flash("you are successfully logged in")

    return render_template('ShopkeeperHome.html', data=data)

@app.route('/ShopkeeperHome')

def ShopkeeperHome():

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM keepertb where username='" + session['uname']
    + "'")

    data = cur.fetchall()

    return render_template('UserHome.html', data=data)

@app.route("/userlogin", methods=['GET', 'POST'])

def userlogin():

    if request.method == 'POST':

        username = request.form['uname']

```

```

password = request.form['password']

session['uname'] = request.form['uname']

conn = mysql.connector.connect(user='root', password='', host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

cursor.execute("SELECT * from regtb where username='" + username + "' and
password='" + password + "'")

data = cursor.fetchone()

if data is None:

    flash('Username or Password is wrong')

    return render_template('UserLogin.html', data=data)

else:

    mob = data[2]

    email = data[3]

    import random

    n = random.randint(1111, 9999)

    sendmsg(mob, "Your OTP" + str(n))

    sendmail(email, "Your OTP" + str(n))

    session['otp'] = str(n)

    # flash("you are successfully logged in")

    return render_template('OTP.html', data=data)

@app.route("/otplogin", methods=['GET', 'POST'])

def otplogin():

    error = None

    if request.method == 'POST':

```

```

username = request.form['ot']

if session['otp'] == username:

    username1 = session['uname']

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM regtb where username='" + username1 + "'")

    data = cur.fetchall()

    flash("you are successfully logged in")

    return render_template('UserHome.html', data=data)

else:

    flash('OTP is Incorrect!')

    return render_template('OTP.html')

@app.route('/UserHome')

def UserHome():

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM regtb where username='" + session['uname'] +
    "' ")

    data = cur.fetchall()

    return render_template('UserHome.html', data=data)

@app.route('/Search')

def Search():

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

```

```

cur = conn.cursor()

cur.execute("SELECT * FROM protb ")

data = cur.fetchall()

return render_template('Search.html', data=data)

@app.route("/csearch", methods=['GET', 'POST'])

def csearch():

    if request.method == 'POST':

        ptype = request.form['ptype']

        conn = mysql.connector.connect(user='root', password="", host='localhost',
        database='25onlinemobiledb')

        cur = conn.cursor()

        cur.execute(

            "SELECT * FROM protb where ProductType='" + ptype + "'" )

        data = cur.fetchall()

        return render_template('Search.html', data=data)

@app.route("/add")

def add():

    flash("Please Login")

    return render_template('index.html')

@app.route("/Add")

def Add():

    id = request.args.get('id')

    session['pid'] = id

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

```

```

cursor = conn.cursor()

cursor.execute("SELECT * FROM protb where id='" + id + "'")

data = cursor.fetchone()

if data:

    name = data[1]

    desc = data[6]

    amount = data[5]

    image = data[7]

else:

    return 'Incorrect username / password !'

return      render_template('FullInfo.html',      name=name,      desc=desc,
amount=amount, image=image)

@app.route("/addcart", methods=['GET', 'POST'])

def addcart():

    if request.method == 'POST':

        import datetime

        date = datetime.datetime.now().strftime('%Y-%m-%d')

        pid = session['pid']

        uname = session['uname']

        qty = request.form['qty']

        # pmode = request.form['pmode']

        conn = mysql.connector.connect(user='root', password="", host='localhost',
        database='25onlinemobiledb')

        cursor = conn.cursor()

        cursor.execute("SELECT * FROM protb where id='" + str(pid) + "'")

```

```

data = cursor.fetchone()

if data:

    ProductName = data[1]

    Producttype = data[2]

    price = data[5]

    cQty = data[8]

    Image = data[7]

else:

    return 'No Record Found!'

    tprice = float(price) * float(qty)

    clqty = float(cQty) - float(qty)

    if clqty < 0:

        flash('Low Product ')

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM protb where id=" + pid + " ")

    data = cur.fetchall()

    return render_template('AddCart.html', data=data)

else:

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cursor = conn.cursor()

    cursor.execute(

        "SELECT count(*) As count FROM booktb ")

```

```

data = cursor.fetchone()

if data:

    bookno = data[0]

    print(bookno)

    if bookno == 'Null' or bookno == 0:

        bookno = 1

    else:

        bookno += 1

    else:

        return 'Incorrect username / password !'

    bookno = 'BOOKID' + str(bookno)

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cursor = conn.cursor()

    cursor.execute(

        "INSERT INTO carttb VALUES ('" + uname + "','" + ProductName + "','" +
        Producttype + "','" + str(
        price) + "','" + str(qty) + "','" + str(tprice) + "','" +
        Image + "','" + date + "','0','" + bookno + "')"

    conn.commit()

    conn.close()

    flash('Product Add to Card Successfully')

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

```

```

cur.execute("SELECT * FROM protb ")

data = cur.fetchall()

return render_template('Search.html', data=data)

@app.route("/Cart")

def Cart():

    uname = session['uname']

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cur = conn.cursor()

    cur.execute("SELECT * FROM carttb where UserName='" + uname + "' and
    Status='0' ")

    data = cur.fetchall()

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cursor = conn.cursor()

    cursor.execute(

    "SELECT sum(Qty) as qty ,sum(Tprice) as Tprice FROM carttb where
    UserName='" + uname + "' and Status='0' ")

    data1 = cursor.fetchone()

    if data1:

        tqty = data1[0]

        tprice = data1[1]

    else:

        return 'No Record Found!'

    return render_template('Cart.html', data=data, tqty=tqty, tprice=tprice)

@app.route("/RemoveCart")

```



```

def RemoveCart():

id = request.args.get('id')

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor= conn.cursor()

cursor.execute(

"delete from carttb where id='" + id + "'")

conn.commit()

conn.close()

flash('Product Remove Successfully!')

uname = session['uname']

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM carttb where UserName='" + uname + "' and
Status='0' ")

data = cur.fetchall()

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor= conn.cursor()

cursor.execute(

"SELECT sum(Qty) as qty ,sum(Tprice) as Tprice FROM carttb where
UserName='" + uname + "' and Status='0' ")

data1 = cursor.fetchone()

if data1:

tqty = data1[0]

```

```

tprice = data1[1]

return render_template('Cart.html', data=data, tqty=tqty, tprice=tprice)

def loginvalet1():

uname = session['uname']

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

cursor.execute("SELECT * FROM regtb where UserName='" + uname + "'")

data = cursor.fetchone()

if data:

Email = data[3]

Phone = data[2]

else:

return 'Incorrect username / password !'

return uname, Email, Phone

@app.route("/payment", methods=['GET', 'POST'])

def payment():

if request.method == 'POST':

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

cursor.execute("truncate table temptb")

conn.commit()

conn.close()

try:

```

```

import MyNewCode as liv1

del sys.modules["MyNewCode"]

except OSError as e:

print("Caught OSError:", e)

uname = session['uname']

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

cursor.execute("SELECT * from temptb where UserName='" + uname + "'")

data = cursor.fetchone()

if data is None:

flash('Face is wrong')

return render_template('Cart.html')

else:

import datetime

date = datetime.datetime.now().strftime('%Y-%m-%d')

uname = session['uname']

cname = request.form['cname']

Cardno = request.form['cno']

Cvno = request.form['cvno']

tqty=0

tprice=0

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

```

```

cursor.execute(
    "SELECT sum(Qty) as qty ,sum(Tprice) as Tprice FROM carttb where
    UserName='" + uname + "' and Status='0' ")

data1 = cursor.fetchone()

if data1:

    tqty = data1[0]

    tprice = data1[1]

    conn = mysql.connector.connect(user='root', password="", host='localhost',
    database='25onlinemobiledb')

    cursor = conn.cursor()

    cursor.execute(
        "SELECT count(*) As count FROM booktb ")

    data = cursor.fetchone()

    if data:

        bookno = data[0]

        print(bookno)

        if bookno == 'Null' or bookno == 0:

            bookno = 1

        else:

            bookno += 1

        else:

            return 'Incorrect username / password !'

        bookno = 'BOOKID' + str(bookno)

        conn = mysql.connector.connect(user='root', password="", host='localhost',
        database='25onlinemobiledb')

```

```

cursor=conn.cursor()

cursor.execute(

"update   carttb set status='1',Bookid='" + bookno + "' where UserName='" +
uname + "' ")

conn.commit()

conn.close()

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor=conn.cursor()

cursor.execute(

"INSERT INTO booktb VALUES ('" + uname + "','" + bookno + "','" + str(tqty)
+ "','" + str(

tprice) + "','" + cname + "','" + Cardno + "','" + Cvno + "','" + date + "')"

conn.commit()

conn.close()

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM   carttb where UserName='" + uname + "' and
Status='1' ")

data1 = cur.fetchall()

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM booktb where username='" + uname + "'")

data2 = cur.fetchall()

```

```

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cursor = conn.cursor()

cursor.execute("SELECT * from regtb where username='" + uname + "'")

data33 = cursor.fetchone()

if data33:

mob = data33[2]

sendmsg(mob, 'Product On the Way')

flash('Payment Successfully..!')

return render_template('UserBook.html', data1=data1, data2=data2)

@app.route("/BookInfo")

def BookInfo():

uname = session['uname']

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM carttb where UserName='" + uname + "' and
Status='1' ")

data1 = cur.fetchall()

conn = mysql.connector.connect(user='root', password="", host='localhost',
database='25onlinemobiledb')

cur = conn.cursor()

cur.execute("SELECT * FROM booktb where username='" + uname + "'")

data2 = cur.fetchall()

return render_template('UserBook.html', data1=data1, data2=data2)

def sendmsg(targetno, message):

```

```

import requests

requests.post(
    "http://sms.creativepoint.in/api/push.json?apikey=6555c521622c1&route=transs
ms&sender=FSSMSS&mobilenos=" + targetno + "&text=Dear customer your
msg is " + message + " Sent By FSMSG FSSMSS")

def sendmail(Mailid, message):

    import smtplib

    from email.mime.multipart import MIMEMultipart

    from email.mime.text import MIMEText

    from email.mime.base import MIMEBase

    from email import encoders

    fromaddr = "projectmailm@gmail.com"

    toaddr = Mailid

    # instance of MIMEMultipart

    msg = MIMEMultipart()

    # storing the senders email address

    msg['From'] = fromaddr

    # storing the receivers email address

    msg['To'] = toaddr

    # storing the subject

    msg['Subject'] = "Alert"

    # string to store the body of the mail

    body = message

    # attach the body with the msg instance

    msg.attach(MIMEText(body, 'plain'))

```

```
# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login(fromaddr, "qmgn xecl bkqv musr")

# Converts the Multipart msg into a string
text = msg.as_string()

# sending the mail
s.sendmail(fromaddr, toaddr, text)

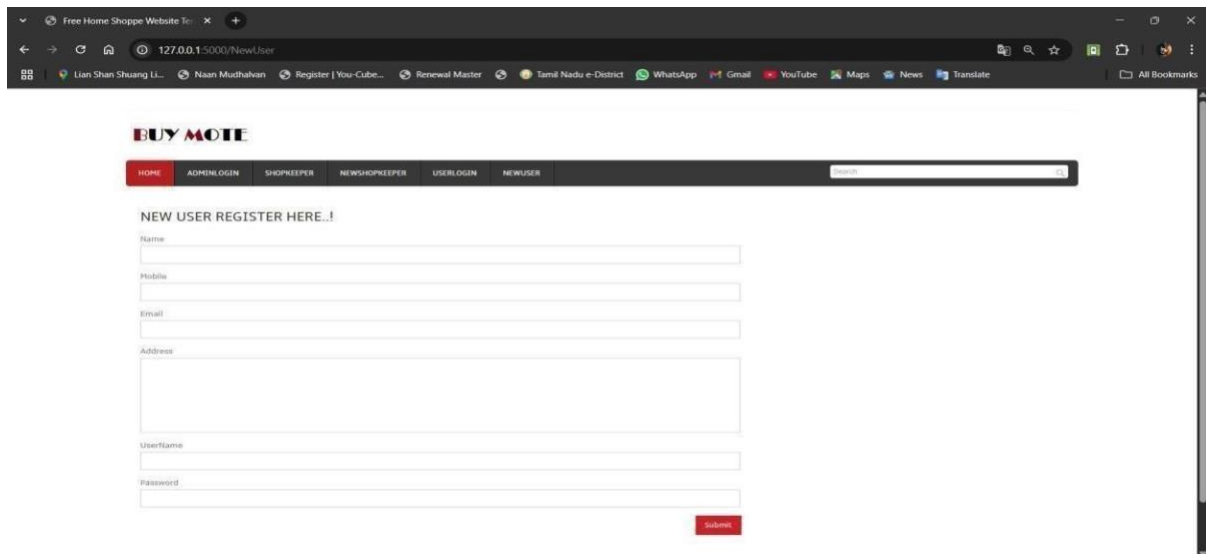
# terminating the session
s.quit()

if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```



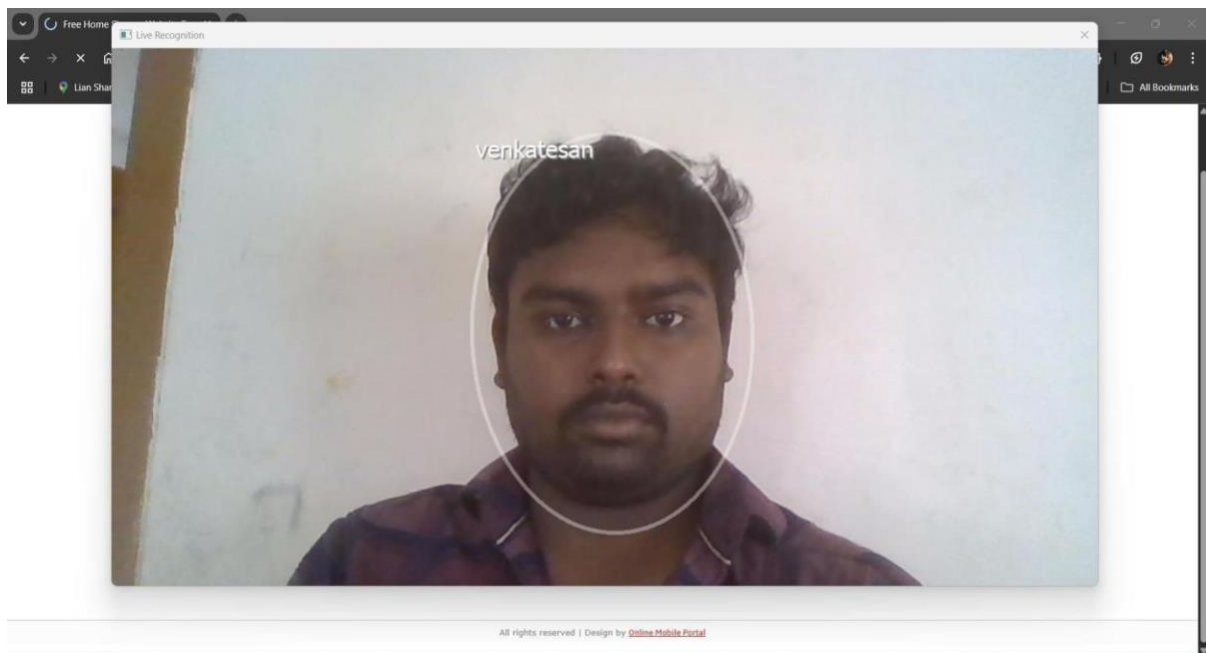
## APPENDIX II

### 8.2 SCREENSHOTS



The screenshot displays a web browser window with the address bar showing '127.0.0.1:5000/NewUser'. The website header includes the 'BUY MOTE' logo and a navigation menu with links: HOME, ADMIN LOGIN, SHOPKEEPER, NEWSHOPKEEPER, USER LOGIN, and NEWUSER. A search bar is located to the right of the navigation menu. The main content area features a registration form titled 'NEW USER REGISTER HERE...!'. The form contains input fields for Name, Mobile, Email, Address, Username, and Password, followed by a red 'Submit' button.

#### A.2.1 New User Registration



#### A.2.2 Biometric by face Verification

**BUY MOTE**

HOME ADMINLOGIN SHOPKEEPER NEWSHOPKEEPER USERLOGIN NEWUSER

Search

**USER LOGIN HERE..!**

Username

Password

Login

All rights reserved | Design by [Online Mobile Portal](#)

## A.2.3 User Login

**BUY MOTE**

HOME ADMINLOGIN SHOPKEEPER NEWSHOPKEEPER USERLOGIN NEWUSER

Search

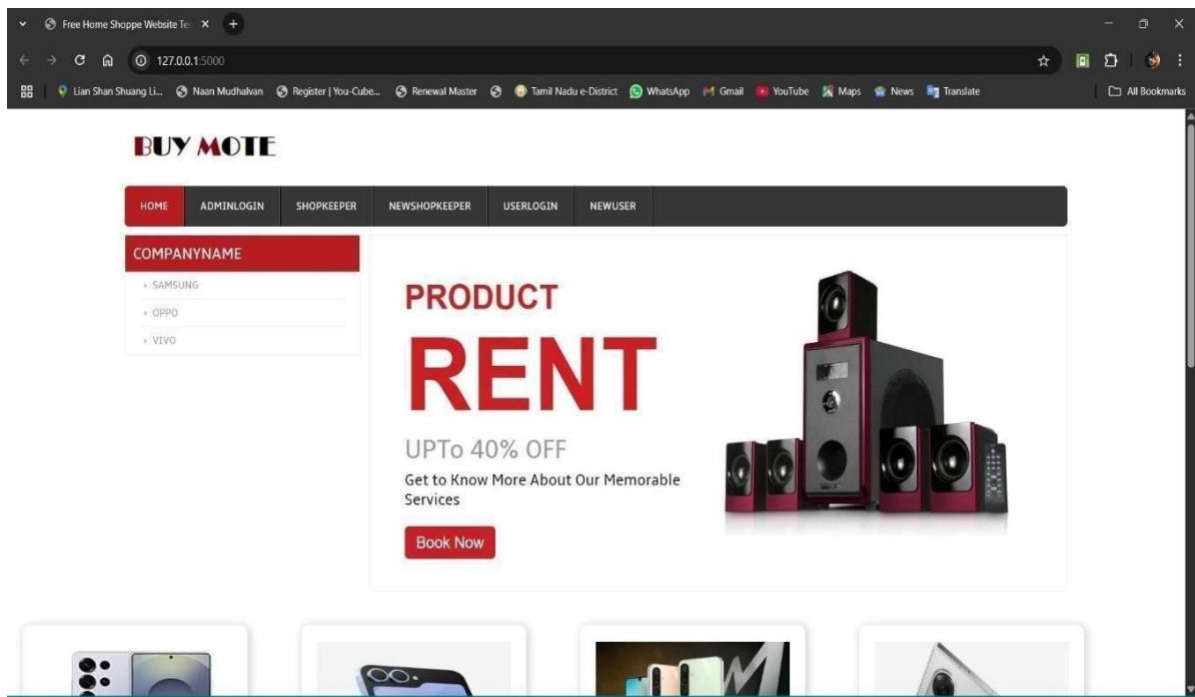
**OTP VERIFICATION HERE..!**

Enter OTP

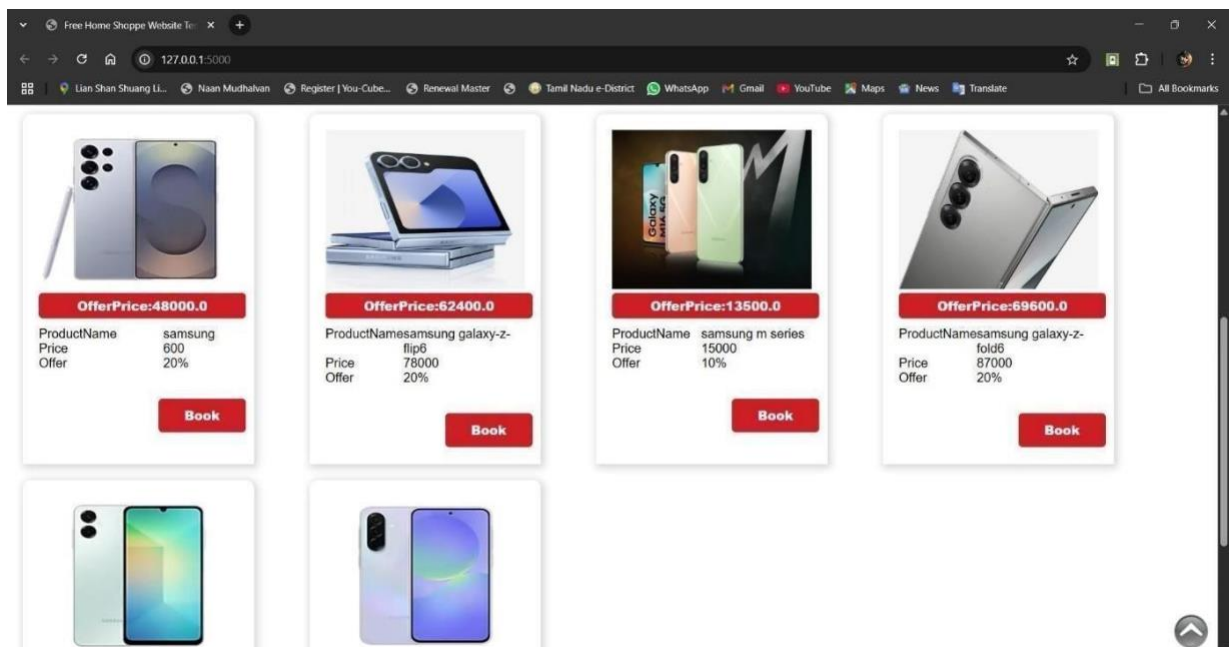
Login

All rights reserved | Design by [Online Mobile Portal](#)

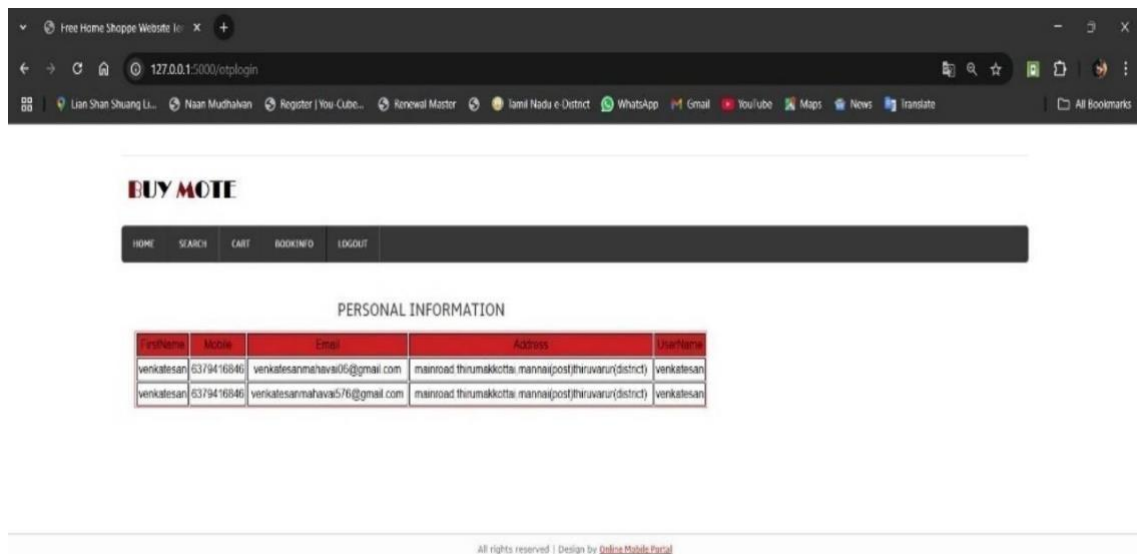
## A.2.4 OTP Verification



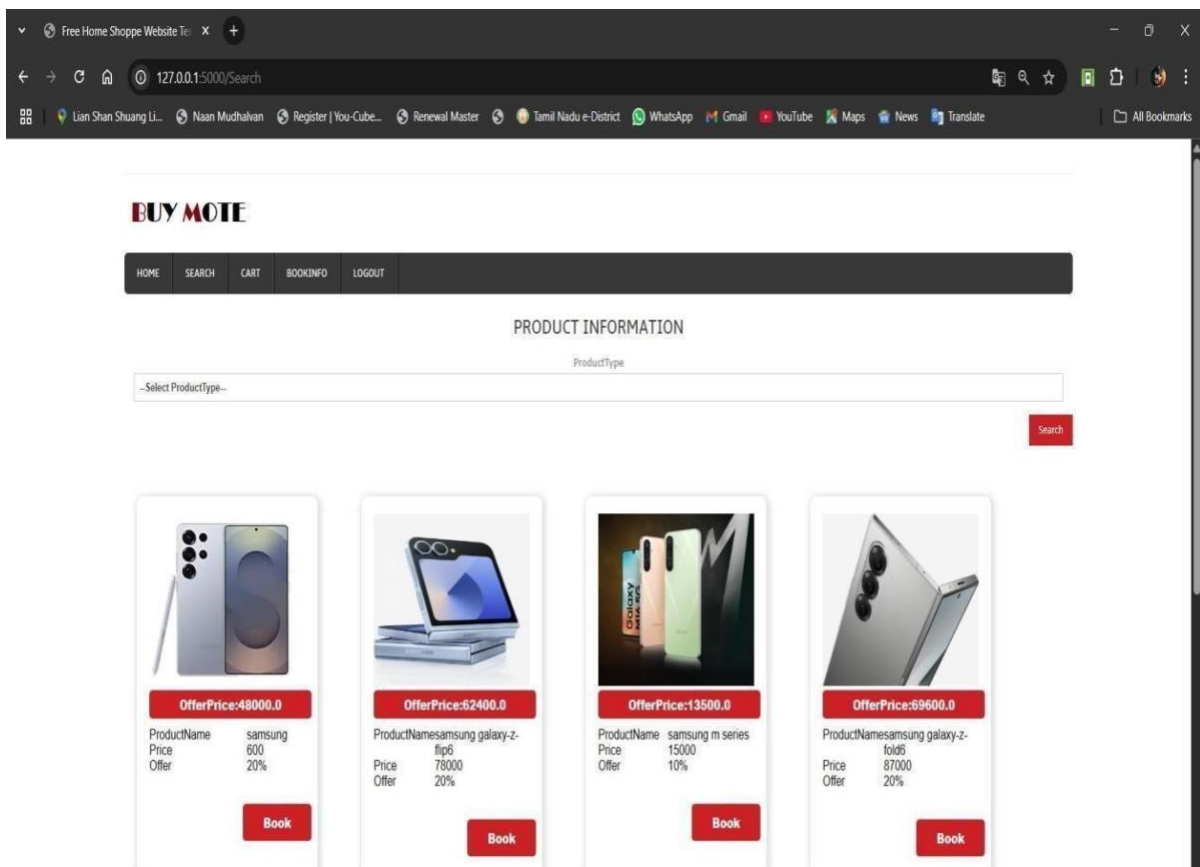
## A.2.5 Home Page



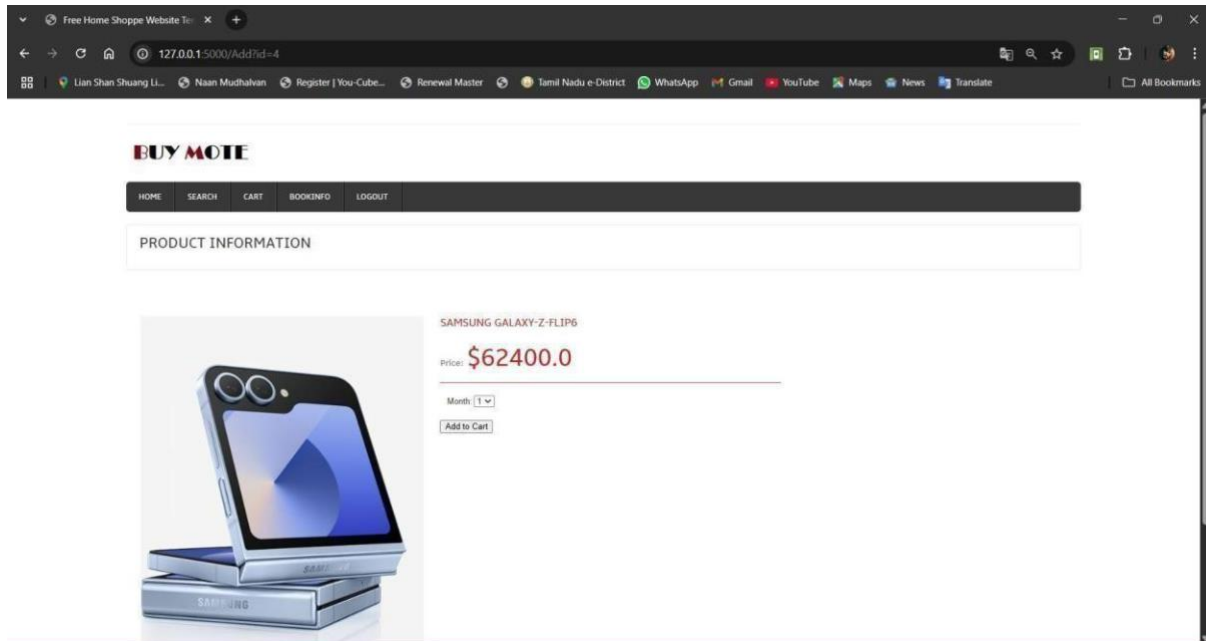
## A.2.6 Product Purchase



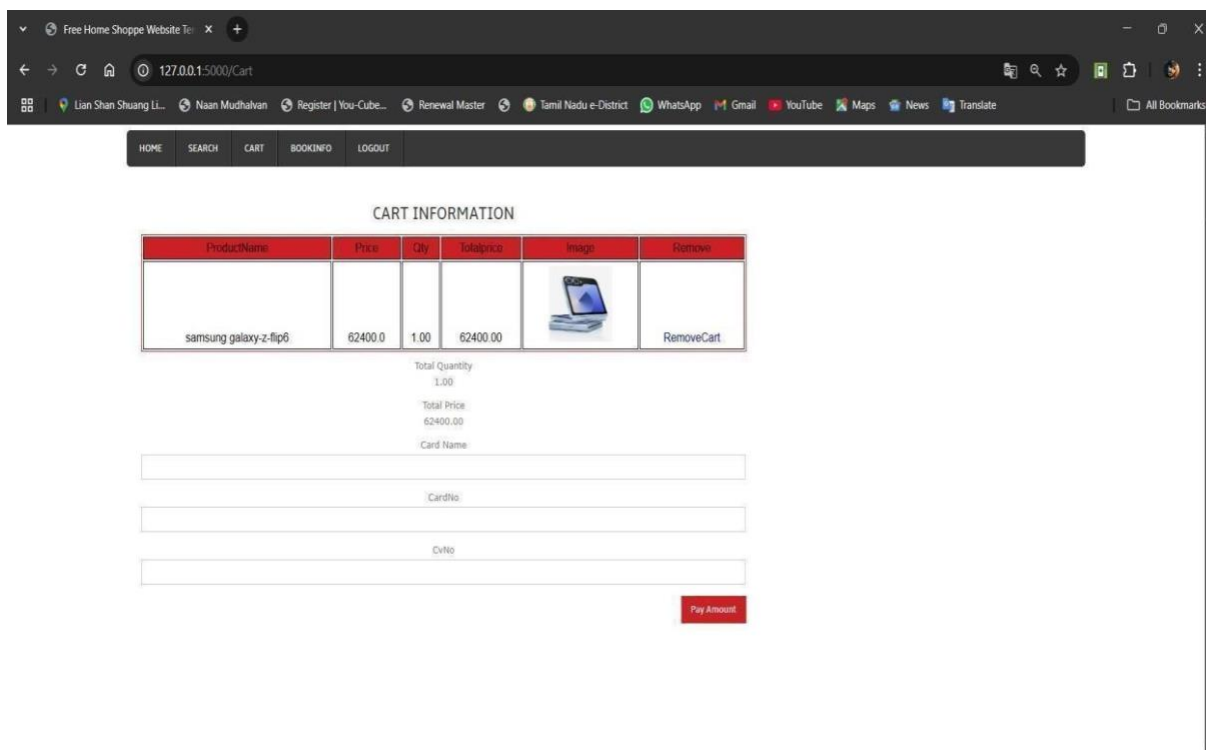
## A.2.7 Personal Information



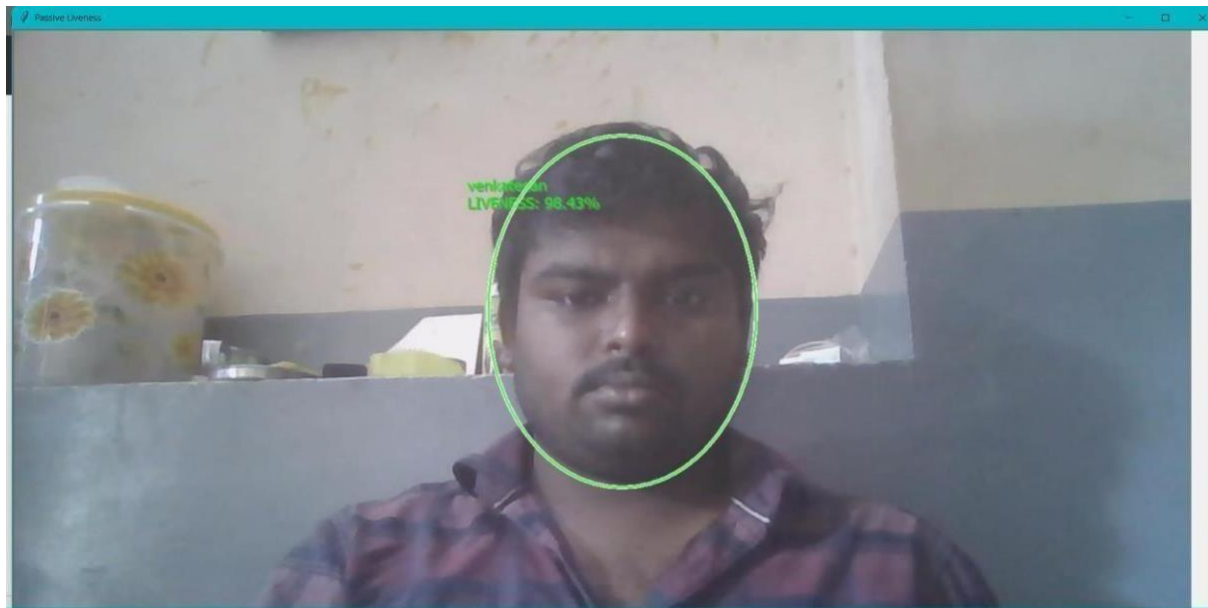
## A.2.8 Product Information



## A.2.9 Product Confirmation



## A.2.10 Cart Information



### A.2.11 Face Recognition by Liveness



Free Home Shoppe Website Te... x +

127.0.0.1:5000/payment

BUY MOTE

HOME SEARCH CART BOOKINFO LOGOUT

PRODUCT INFORMATION

ProductName	Price	Qty	Totalprice	Image	Book id
samsung	784000.00	1.00	784000.00		BOOKID6
samsung galaxy-z-flip6	62400.00	1.00	62400.00		BOOKID6

PAYMENT INFORMATION

UserName	Book id	Qty	Amount	CertName	Date
venkatesan	BOOKID6	1.00	784000.00	mastercard	2025-04-16
venkatesan	BOOKID6	1.00	62400.00	mastercard	2025-04-29

### A.2.12 Product & Payment Information

**BUY MOTE**

HOME ADMINLOGIN SHOPKEEPER NEWSHOPKEEPER USERLOGIN NEWUSER

ADMINLOGIN

UserName

Password

Login

All rights reserved | Design by [Online Mobile Portal](#)

## A.2.13 Admin Login

**BUY MOTE**

HOME PRODUCTINFO SALESINFO LOGOUT

USER INFORMATION

First Name	Mobile	Email	Address	User Name	Password
venkat	6379416846	venkatesanmahava576@gmail.com	n0	venkat	venkat
venkatesan	6379416846	venkatesanmahava06@gmail.com	mainroad,thirumakkottai,manai(post)thiruvarur(district)	venkatesan	venkatesan
mahavai	6379416846	venkatesanmahava576@gmail.com	mainroad,thirumakkottai,manai(post)thiruvarur(district)	mahavai	mahavai
ram	9047858596	chandran1989@gmail.com	ok	ram	ram
vignesh	7568258550	e1215115vignesh@gmail.com	hict b gh	vicky	0123
venkatesan	6379416846	venkatesanmahava576@gmail.com	mainroad,thirumakkottai,manai(post)thiruvarur(district)	venkatesan	venkatesan

All rights reserved | Design by [Online Mobile Portal](#)

## A.2.14 User Information

Free Home Shoppe Website T... x +






127.0.0.1:5000/ProductInfo

Lian Shan Shuang Li... Naan Mudhalvan Register | You-Cube... Renewal Master Tamil Nadu e-District WhatsApp Gmail YouTube Maps News Translate All Bookmarks

**BUY MOTE**

HOME PRODUCTINFO SALEINFO LOGOUT

PRODUCT INFORMATION

ProductName	Type	Price	offer	Amount	Image	Remove
samsung	Mobile	600	20	48000.0		Remove
samsung galaxy-z-flip6	Mobile	78000	20	62400.0		Remove
samsung m series	Mobile	15000	10	13500.0		Remove
samsung galaxy-z-fold6	Mobile	87000	20	69600.0		Remove
samsung A06	Mobile	20000	10	18000.0		Remove

## A.2.15 Product Information

Free Home Shoppe Website T... x +






127.0.0.1:5000/SaleInfo

Lian Shan Shuang Li... Naan Mudhalvan Register | You-Cube... Renewal Master Tamil Nadu e-District WhatsApp Gmail YouTube Maps News Translate All Bookmarks

**BUY MOTE**




HOME NEWPRODUCT PRODUCTINFO SALEINFO LOGOUT

APPROVED WAITING RENT INFORMATION

ProductName	Price	Qty	Sub-price	Image	Book id
samsung	784000.0	1.00	784000.00		BOOKID6
samsung	784000.0	1.00	784000.00		BOOKID5
samsung	48000.0	1.00	48000.00		BOOKID3
samsung galaxy-z-flip6	62400.0	1.00	62400.00		BOOKID4
					

## A.2.16 Approved for Rent Information



samsung galaxy-z-flip6	62400.0	1.00	62400.00		BOOKID4
samsung A36	29750.0	1.00	29750.00		BOOKID5
samsung galaxy-z-flip6	62400.0	1.00	62400.00		BOOKID6

PAYMENT INFORMATION

UserName	Book id	Qty	Amount	CardName	Date
venkatesan	BOOKID6	1.00	784000.00	mastercard	2025-04-16
mahavai	BOOKID7	1.00	784000.00	rupay	2025-04-16
ram	BOOKID3	1.00	48000.00	mastercard	2025-04-22
vicky	BOOKID4	1.00	62400.00	rupay	2025-04-22
mahavai	BOOKID5	1.00	29750.00	mastercard	2025-04-25
venkatesan	BOOKID6	1.00	62400.00	mastercard	2025-04-29

All rights reserved | Design by [Online Mobile Portal](#)

## A.2.17 Payment Information

HOME	ADMINLOGIN	SHOPKEEPER	NEWSHOPKEEPER	USERLOGIN	NEWUSER	<input type="text" value="Search"/>
------	------------	------------	---------------	-----------	---------	-------------------------------------

NEW SHOP KEEPER REGISTER HERE..!

Name

Mobile

Email

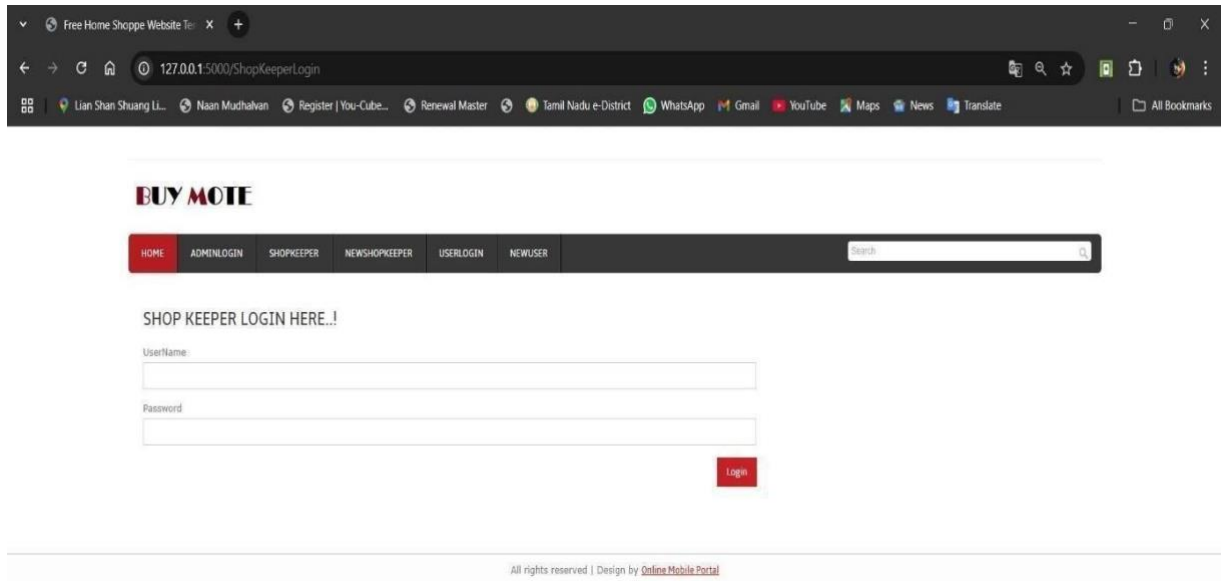
Address

Username

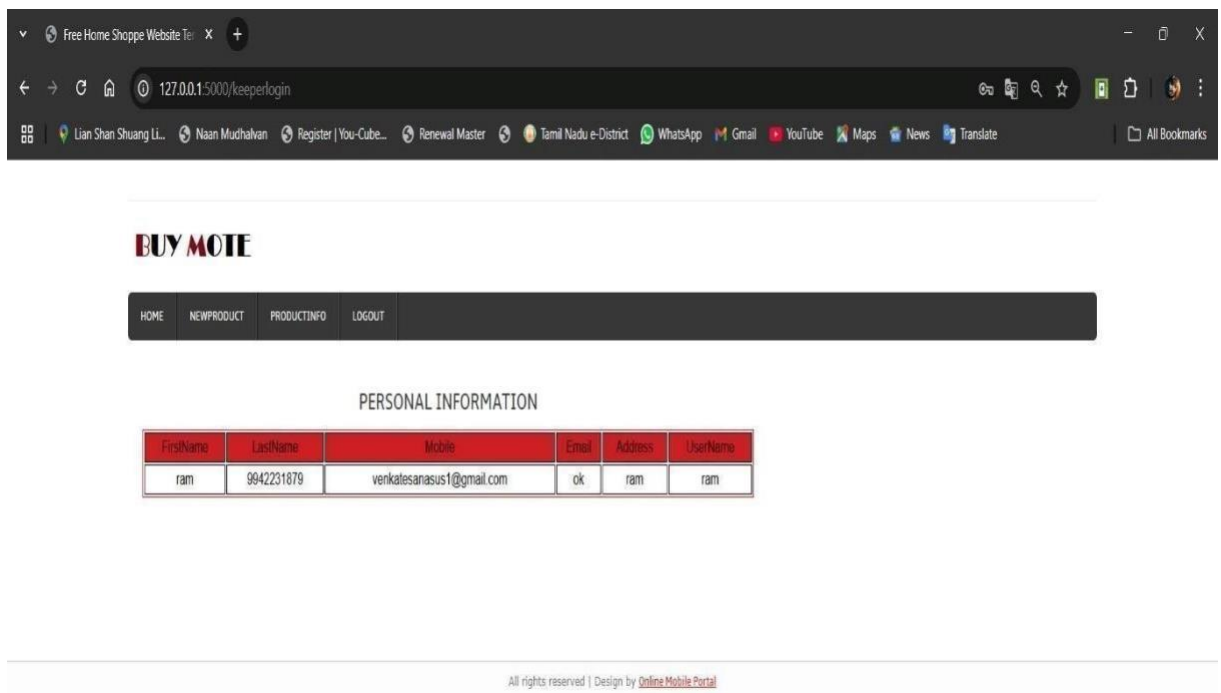
Password

All rights reserved | Design by [Online Mobile Portal](#)

## A.2.18 New Shop Keeper Registration



## A.2.19 Shop Keeper Login Page



## A.2.20 Personal Information

**BUY MOTE**

HOME NEWPRODUCT PRODUCTINFO LOGOUT

NEW PRODUCT REGISTER HERE..!

Name:

ProductType:

Stock:

Amount:

offer%:

Info:







Image:  No file chosen

## A.2.21 New Product Registration

**BUY MOTE**

HOME NEWPRODUCT PRODUCTINFO LOGOUT

PRODUCT INFORMATION

ProductName	Type	Price	offer	Amount	Image	Remove
samsung	Mobile	600	20	48000.0		<a href="#">Remove</a>
samsung galaxy-z-flip6	Mobile	78000	20	62400.0		<a href="#">Remove</a>
samsung m series	Mobile	15000	10	13500.0		<a href="#">Remove</a>
samsung galaxy-z-fold6	Mobile	87000	20	69600.0		<a href="#">Remove</a>
samsung A06	Mobile	20000	10	18000.0		<a href="#">Remove</a>
						

## A.2.22 Product Information

## CHAPTER 9

### REFERENCES

1. Sahu, Aanchal, G. M. Harshvardhan, and Mahendra Kumar Gourisaria. "A dual approach for credit card fraud detection using neural network and data mining techniques." In 2020 IEEE 17th India council international conference (INDICON), pp. 1-7. IEEE, 2020.
2. Panda, Agyan, BharathYadlapalli, and Zhi Zhou. "Credit card fraud detection through machine learning algorithm." Big Data and Computing Visions 1, no. 3 (2021): 140-145.
3. Aziz, Amir, and Hamid Ghous. "Fraudulent Transactions Detection in Credit Card by using Data Mining Methods: A Review." INTERNATIONAL JOURNAL OF SCIENTIFIC PROGRESS AND RESEARCH (IJSPR) 79, no. 179 (2021).
4. Shah, Ankit, and Akash Mehta. "Comparative Study of Machine Learning Based Classification Techniques for Credit Card Fraud Detection." In 2021 International Conference on Data Analytics for Business and Industry (ICDABI), pp. 53-59. IEEE, 2021.
5. Mienye, IbomoiyeDomor, and NobertJere. "Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions." *IEEE Access* (2024).
6. Singh, Gurpreet, DivyanshiKaushik, HritikHanda, GagandeepKaur, Sunil Kumar Chawla, and A. Ahmed. "BioPay: a secure payment gateway through biometrics." Journal of Cybersecurity and Information Management 7, no. 2 (2021): 65-76.

7. Tiwari, Pooja, Simran Mehta, NishthaSakhuja, Jitendra Kumar, and Ashutosh Kumar Singh. "Credit card fraud detection using machine learning: a study." arXiv preprint arXiv:2108.10005 (2021).
8. Kumar, Sheo, Vinit Kumar Gunjan, MohdDilshad Ansari, and RashmiPathak. "Credit Card Fraud Detection Using Support Vector Machine." In Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications: ICMISC 2021, pp. 27-37. Springer Singapore, 2022.
9. Vinaya, D. S., Satish B. Basapur, VanishreeAbhay, and NeethaNatesh. "Credit Card Fraud Detection Systems (CCFDS) using Machine Learning (Apache Spark)." (2020).
10. Lucas, Yvan, and Johannes Jurgovsky. "Credit card fraud detection using machine learning: A survey." arXiv preprint arXiv:2010.06479 (2020).
11. Singh, D. and Singh, M., Investigation of OpenCV for real-time face detection using Haar features. International Journal of Computer Applications, 975:8887 (2020).
12. Hjeltnæs, E. and Low, B.K., Face detection: A survey. Computer Vision and Image Understanding, 83(3), pp.236–274. (2001).
13. Hamm, J. and Lee, D.D., Grassmann discriminant analysis: a unifying view on subspace-based learning. In Advances in Neural Information Processing Systems, 21. (2008).
14. Jain, A.K., Ross, A. and Nandakumar, K., Introduction to biometrics. Springer Science & Business Media. ISBN: 978-0-387-77326-1 (2011).

## CERTIFICATES



**SHIVANI ENGINEERING COLLEGE**  
(A Unit of Shivani Institutions)  
(NAAC ACCREDITED)  
(Approved by AICTE, New Delhi & Affiliated to Anna University Chennai.)  
Ammapettai Village, Poolangulathupatti (PO), Sriirangam Taluk, Trichy- 620009.  
Mob : 9750965056 / 94866 46392 Email : neest2025@shivani.ac.in



### Certificate of Participation

This to Certify that Dr. / Mr./ Mrs. / Ms. ....VENKATESAN.A...../CSE.CCS).....

.....M.T.E.T.....ENGINEERING.....COLLEGE.....TRICHY.....

has participated & presented a paper entitled ..REAL-TIME...CREDIT...CARD.....

.....FRAUD...DETECTION.....

in the National Conference on "Emerging trends in Engineering, Science and Technology  
(NEEST 25)" on 16<sup>th</sup> May 2025.



Co-Ordinator



Convenor



Principal



# SHIVANI ENGINEERING COLLEGE

(A Unit of Shivani Institutions)

(NAAC ACCREDITED)

(Approved by AICTE, New Delhi & Affiliated to Anna University Chennai.)

Ammapettai Village, Poolangulathupatti (PO), Srirangam Taluk, Trichy- 620009.

Mob : 9750965056 / 94866 46392 Email : neest2025@shivani.ac.in



## Certificate of Participation

This to Certify that Dr. / Mr./ Mrs. / Ms. ....VIGNESH.....A.....I.C.S.E.C.S.).....

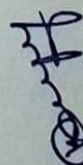
.....M.T.E.T.....ENGINEERING.....COLLEGE.....


has participated & presented a paper entitled .REAL...TIME...CREAT....CARA.....

.....FRAUD.....DETECTION.....

in the National Conference on "Emerging trends in Engineering, Science and Technology

(NEEST 25)" on 16<sup>th</sup> May 2025.

  
Co-Ordinator

  
Convenor

  
Principal



# SHIVANI ENGINEERING COLLEGE

(A Unit of Shivani Institutions)

(NAAC ACCREDITED)

(Approved by AICTE, New Delhi & Affiliated to Anna University Chennai.)

Ammapettai Village, Poolangulathupatti (PO), Srirangam Taluk, Trichy- 620009.

Mob : 9750965056 / 94866 46392 Email : neest2025@shivani.ac.in



## Certificate of Participation

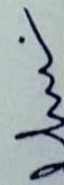
This to Certify that Dr. / Mr./ Mrs. / Ms. ....AJAY: A...../CSE(CS).....

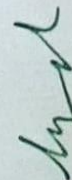
.....M.T.E.T.....ENGINEERING.....COLLEGE.....


has participated & presented a paper entitled REAL TIME CREDIT CARD.....

.....FRAUD.....DETECTION.....

in the National Conference on "Emerging trends in Engineering, Science and Technology  
(NEEST 25)" on 16<sup>th</sup> May 2025.

  
Co-Ordinator

  
Convenor

  
Principal



# SHIVANI ENGINEERING COLLEGE

(A Unit of Shivani Institutions)

(NAAC ACCREDITED)

(Approved by AICTE, New Delhi & Affiliated to Anna University Chennai.)

Ammapettai Village, Poolangulathupatti (PO), Srirangam Taluk, Trichy- 620009.

Mob : 9750965056 / 94866 46392 Email : neest2025@shivani.ac.in



## Certificate of Participation

This to Certify that Dr. / Mr./ Mrs. / Ms. ...ELIAS...N...../...CSE.C.S.S).....

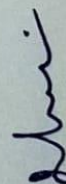
.....M.I.E.T.....ENGINEERING.....COLLEGE.....

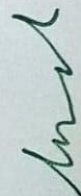
has participated & presented a paper entitled ..REAL TIME...CREDIT...CARD.....

.....FRAUD.....DETECTION.....

in the National Conference on "Emerging trends in Engineering, Science and Technology

(NEEST 25)" on 16<sup>th</sup> May 2025.

  
Co-Ordinator

  
Convenor

  
Principal