

FINGER PRINT BASED STUDENT ATTENDENCE MANAGEMENT

A project submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Science in Computer Science to the
Mahendra Arts & Science College (Autonomous)**

by

SANTHOSH.V

(21BCS1051)

Under the Guidance of

Mrs. P.MALATHI, M.C.A., M.Phil.,



DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS

MAHENDRA ARTS & SCIENCE COLLEGE

(AUTONOMOUS)

(Accredited by NAAC with „A++“Grade & Affiliated to Periyar University)

Kalippatti (PO) – 637 501, Namakal (DT).

APRIL / MAY-2024

MAHENDRA ARTS & SCIENCE COLLEGE (AUTONOMOUS)

Kalippatti

(Affiliated to Periyar University, Salem)



This is to justify that the project entitled

FINGER PRINT BASED STUDENT ATTENDANCE MANAGEMENT

is the bonafide record of project work done

by

SANTHOSH V

(21BCS1051)

A project submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science to the

Mahendra Arts & Science College (Autonomous)

Mrs. P.MALATHI .M.C.A., M.Phil.,

Assistant Professor,

Dept. of Computer Science & Applications,

Mahendra Arts & Science College (Autonomous),

Kalippatti-637 501.

Mrs. M. SUMATHI, M.Sc., M.Phil., B.Ed.,

Head of the Department,

Dept. of Computer Science & Applications,

Mahendra Arts & Science College (Autonomous),

Kalippatti-637 501.

Submitted for viva-voce examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **SANTHOSH V** hereby declare that the project work, entitled “ **FINGER PRINT BASED STUDENT ATTENDENCE MANAGEMENT** ” submitted to the Mahendra Arts & Science College (Autonomous), Kalippatti in partial fulfillment of the requirements for the award of the degree of **Bachelor of Computer Science** is a record of the original project work done by me under the supervision and guidance of **Mrs. P.MALATHI., M.C.A., M.Phil., Assistant Professor**, Department of Computer Science & Applications, Mahendra Arts & Science College (Autonomous), Kalippatti and it has not formed the basis for the award of any Degree / Diploma / Associate ship / Fellowship or other similar title to any candidate in any university.

Place: Kalippatti

Signature of the Candidate

Date: 21.05.2024

[SANTHOSH V]

ACKNOWLEDGEMENT

My sincere gratitude to **Sri. M. G. BHARATHKUMAR, M.A., B.Ed., Chairman** of Mahendra Educational Trust for offering me an opportunity and providing me all the facilities to do my project.

I sincerely thank the **Secretary** of Mahendra Educational Trust Smt. **B. VALLIYAMMAL, M.A., B.Ed.**, for providing excellent facilities.

I extend my sincere thanks to the **Managing Directors** of Mahendra Educational Trust **Mr. B. MAHENDHIRAN** and **Mr. B. MAHA AJAY PRASATH**.

I am grateful to the **Principal** of Mahendra Arts & Science College, Prof. **Dr. S. ARJUNAN, M.Sc., M.Phil., Ph.D., SET.**, for his constant encouragement.

I am grateful to the **Dr. J. JOSEPHINE DAISY, M.Com., M.Phil., MBA., Ph.D.**, the **Controller of Examinations** of Mahendra Arts & Science College for conscientious guidance and encouragement to accomplish this project work.

I express my profound thanks to **Head of the Department** of Computer Science and Applications, **Mrs. M. SUMATHI, M.Sc., M.Phil. B.Ed.**, and for valuable suggestions give us during the project.

I would like to express my deep gratitude to **Mrs. P. MALATHI, M.C.A., M.Phil., Assistant Professor**, Department of Computer Science & Applications for my project guide, for their patient guidance, enthusiastic encouragement and useful critiques of this work.

I would also like to expand my deepest gratitude to all those who have directly and indirectly guided us in writing this assignment work.

I express my deep sense of gratitude to my family members for their whole hearted co-operation and encouragement during the entire study period.

TABLE OF CONTENTS

S.NO	PARTICULARS	PAGE NO
	ABSTRACT	1
1.	INTRODUCTION	2
2.	SYSTEM SPECIFICATION	6
	2.1. HARDWARE SPECIFICATION	6
	2.2. SOFTWARE SPECIFICATION	6
3.	SYSTEM STUDY AND ANALYSIS	7
	3.1. EXISTING SYSTEM	7
	3.2. PROPOSED SYSTEM	8
	3.3. FEASIBILITY STUDY	8
4.	SOFTWARE DESCRIPTION	9
5.	PROJECT DESCRIPTION	16
	5.1. MODULES DESCRIPTION	16
6.	SYSTEM DESIGN AND IMPLEMENTATION	18

7.	SYSTEM TESTING	22
8.	CONCLUSION	24
9.	FEATURE ENHANCEMENT	24
10.	APPENDIX	25
	10.1. SOURCE CODE	25
	10.2. SCREENSHOTS	56
11.	REFERENCES	63

Finger Print Based Student Attendance Management

Abstract

Proper attendance recording and management has become important in today 's world as attendance and achievement go hand in hand. Attendance is one of the work ethics valued by employers. Most of the educational institutions and government organizations in developing countries still use paper based attendance method for maintaining the attendance records. There is a need to replace these traditional methods of attendance recording with biometric attendance system. The unique nature of fingerprint makes it ideal for use in attendance management systems. Besides being secure, Fingerprint based attendance system will also be environment friendly. Fingerprint matching is widely used in forensics for a long time. It can also be used in applications such as identity management and access control. This review incorporates the problems of attendance systems presently in use, working of a typical fingerprint based attendance system, study of different systems, their advantages, disadvantages and comparison based upon important parameters.

1. INTRODUCTION

Biometrics refers to metrics related to human characteristics and traits. Biometrics authentication (or realistic authentication) is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance. Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological versus behavioral characteristics. Physiological characteristics are related to the shape of the body. Examples include, but are not limited to fingerprint, palm veins, fingerprint recognition, DNA, palm print, hand geometry, iris recognition, retina and odour/scent. Behavioral characteristics are related to the pattern of behavior of a person, including but not limited to typing rhythm, gait, and voice. Some researchers have coined the term behavior metrics to describe the latter class of biometrics. More traditional means of access control include token-based identification systems, such as a driver's license or passport, and knowledge-based identification systems, such as a password or personal identification number. Since biometric identifiers are unique to individuals, they are more reliable in verifying identity than token and knowledge-based methods; however, the collection of biometric identifiers raises privacy concerns about the ultimate use of this information.

Biometrics

A number of biometric characteristics may be captured in the first phase of processing. However, automated capturing and automated comparison with previously stored data requires that the biometric characteristics satisfy the following characteristics:

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them. It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
 - Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs. □
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image processing

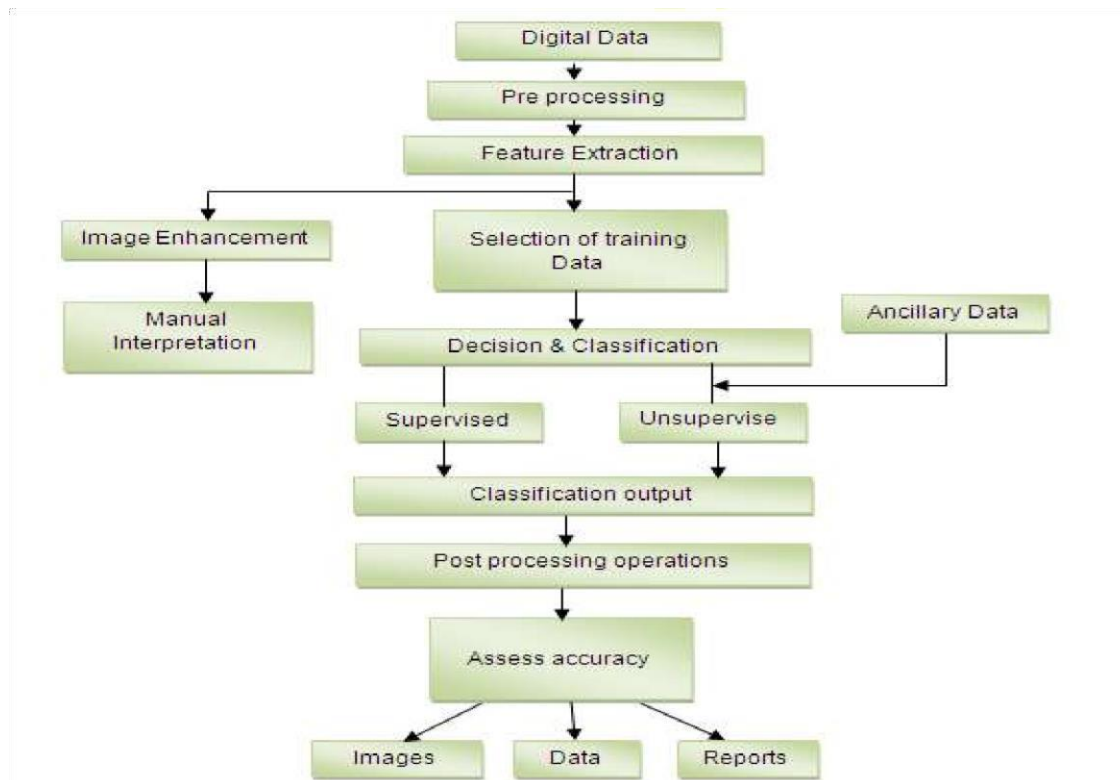
The purpose of image processing is divided into 5 groups. They are:

- Visualization - Observe the objects that are not visible.
- Image sharpening and restoration - To create a better image.
- Image retrieval - Seek for the image of interest.
- Measurement of pattern – Measures various objects in an image.
- Image Recognition – Distinguish the objects in an image.

Types

The two types of methods used for Image Processing are Analog and Digital Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.



In recent years, advances in information technology and telecommunications have acted as catalysts for significant developments in the sector of health care. These technological advances have had a particularly strong impact in the field of medical imaging, where film radiographic techniques are gradually being replaced by digital imaging techniques, and this has provided an impetus to the development of integrated hospital information systems and integrated teleradiology services networks which support the digital transmission, storage, retrieval, analysis, and interpretation of distributed multimedia patient records. One of the many added-value services that can be provided over an integrated teleradiology services network is access to high-performance computing facilities in order to execute computationally intensive image analysis and visualization tasks. In general, currently available products in the field of image processing (IP) meet only specific needs of different end user groups. They either aim to provide a comprehensive pool of ready to use software within a user-friendly and application specific interfingerprnt for those users that use IP software, or aim for the specialised IP researcher and developer, offering programmer's libraries and visual language tools. However, we currently lack the common framework that will integrate all prior efforts and developments in the field and at the same time provide added-value

features that support and in essence realise what we call a „service“. In the case of image processing, these features include: computational resource management and intelligent execution scheduling; intelligent and customizable mechanisms for the description, management, and retrieval of image processing software modules; mechanisms for the “plug-and-play” integration of already existing heterogeneous software modules; easy access and user transparency in terms of software, hardware, and network technologies; sophisticated charging mechanisms based on quality of service; and, methods for the integration with other services available within an integrated health telematics network.

2. SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENTS:

- CPU type : Intel Pentium 4
- Clock speed : 3.0 GHz
- Ram size : 512 MB
- Hard disk capacity : 40 GB
- Monitor type : 15 Inch color monitor
- Keyboard type : internet keyboard

2.2 SOFTWARE REQUIREMENTS:

- Operating System : Windows OS
- Language : ASP .NET
- Back End : SQL SERVER
- IDE : Visual Studio 2012

3. SYSTEM STUDY AND ANALYSIS

3.1 EXISTING SYSTEM

Biometric systems try to identify human beings from their distinctive physiological and behavioral characteristics. Among popular biometric modalities such as iris, fingerprint, voice, hand geometry, and gait, human fingerprints have several advantages which make them attractive for particular applications. Most importantly, the ease of acquiring facial images without the need of subject cooperation allows the use of fingerprint recognition systems in a diverse range of applications such as surveillance systems. However, although recent studies show that the performance of fingerprint recognition systems can reach the level of high security biometric modalities such as fingerprint and iris, it is still a very challenging task to recognize people from their fingerprints under adverse scenarios. Particularly, the presence of illumination differences, in-depth pose variations, and facial expressions are important factors that affect the accuracy of a fingerprint recognition system. With the use of three-dimensional (3-D) facial structure information, it is possible to cope with some of these challenges more efficiently compared to two-dimensional (2-D). In the three dimensional (3-D) domain, challenges caused by illumination, pose, and expression variations can be better handled. However, extreme occlusion variations still complicate the task of identification. In this work, we propose a 3D fingerprint recognition system that is robust under realistic occlusions.

Disadvantages

- Variations caused by pose and expression have attracted increased research effort. □ Not handling occlusions

3.2 PROPOSED SYSTEM

The proposed fingerprint recognition system for students presents an innovative solution to bolster security and streamline various processes within our educational institution. Through a meticulous enrollment process, students' unique fingerprints will be securely stored, forming the basis for a reliable biometric authentication system. This technology will be seamlessly integrated into access control mechanisms for restricted areas, attendance tracking in classrooms and campus events, as well as for library services. By leveraging fingerprint recognition during examinations, we aim to enhance the integrity of the assessment process. The system will not only improve security but also simplify daily transactions, such as cafeteria payments and access to recreational facilities, providing a more convenient and secure campus experience. Prioritizing user-friendly interfaces and robust security measures, our proposal ensures a seamless and efficient adoption of this cutting-edge technology while adhering to privacy regulations and laying the foundation for scalable expansion as our institution evolves.

Advantages

- Improve the security in bio metric applications
- Provide high accuracy
- Can't affected by illumination conditions

3.3 FEASIBILITY STUDY

Feasibility is the determination of whether or not a project is worth doing. The processes is followed in making this determination is called a Feasibility Study. Feasibility study is the test of system proposal according to its workability impact on the organization ability to meet user's needs, and effective use of resources .the result of feasibility study is a formal proposal.

This is simply a report a formal document detailing the nature and scope of the proposed solution. The main objective of a feasibility study is to test the technical, social and economic feasibility of developing a computer system. This is done by investigation the existing system in the area under investigation and generating ideas about a new system

4. SOFTWARE DESCRIPTION:

.NET FRAMEWORK:

The .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

The .NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries. The .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio

Design Features

Interoperability

Because computer systems commonly require interaction between newer and older applications, the .NET Framework provides means to access functionality implemented in newer and older programs that execute outside the .NET environment. Access to COM components is provided in the System.Runtime.InteropServices and System.EnterpriseServices namespaces of the framework; access to other functionality is achieved using the P/Invoke feature.

Common Language Runtime engine

The Common Language Runtime (CLR) serves as the execution engine of the .NET Framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

Language independence

The .NET Framework introduces a Common Type System, or CTS. The CTS specification defines all possible datatypes and programming constructs supported by the CLR and how they may or may not interact with each other conforming to the Common Language Infrastructure (CLI) specification. Because of this feature, the .NET Framework supports the exchange of types and object instances between libraries and applications written using any conforming .NET language.

Base Class Library

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages using the .NET Framework. The BCL provides classes that encapsulate a number of common functions, including file reading and writing, graphic rendering, database interaction, XML document manipulation, and so on. It consists of classes, interfaces of reusable types that integrate with CLR (Common Language Runtime).

Simplified deployment

The .NET Framework includes design features and tools which help manage the installation of computer software to ensure it does not interfere with previously installed software, and it conforms to security requirements.

Security

The design addresses some of the vulnerabilities, such as buffer overflows, which have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

Portability

While Microsoft has never implemented the full framework on any system except Microsoft Windows, it has engineered the framework to be platform-agnostic,^[3] and crossplatform implementations are available for other operating systems (see Silverlight and the Alternative implementations section below). Microsoft submitted the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language), the C# language, and the C++/CLI language^[8] to both ECMA and the ISO, making them available as official standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

Common Language Infrastructure (CLI)

The purpose of the Common Language Infrastructure (CLI) is to provide a language-neutral platform for application development and execution, including functions for Exception handling, Garbage Collection, security, and interoperability. By implementing the core aspects of the .NET Framework within the scope of the CL, this functionality will not be tied to a single language but will be available across the many languages supported by the framework. Microsoft's implementation of the CLI is called the Common Language Runtime, or CLR.

The CIL code is housed in CLI assemblies. As mandated by the specification, assemblies are stored in the Portable Executable (PE) format, common on the Windows platform for all DLL and EXE files. The assembly consists of one or more files, one of which must contain the manifest, which has the metadata for the assembly. The complete name of an assembly (not to be confused with the filename on disk) contains its simple text name, version number, culture, and public key token. Assemblies are considered equivalent if they share the same complete name, excluding the revision of the version number. A private key can also be used by the creator of the assembly for strong naming. The public key token identifies which public key an assembly is signed with. Only the creator of the keypair (typically the .NET developer signing the assembly) can sign assemblies that have the same strong name as a previous version assembly, since he is in possession of the private key. Strong naming is required to add assemblies to the Global Assembly Cache

Security

.NET has its own security mechanism with 2 general features: Code Access Security (CAS), and validation and verification. Code Access Security is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the intranet or Internet). Code Access Security uses evidence to determine the permissions granted to the code. Other code can demand that calling code is granted a specified permission. The demand causes the CLR to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

Class library

Namespaces in the BCL

System

System.Diagnostics

System.Globalization

System.Resources

System.Text

System.Runtime.Serialization

System.Data

The .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built-in APIs are part of either System.* or

Microsoft.* namespaces. These class libraries implement a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET class libraries are available to all CLI compliant languages. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library

The Base Class Library (BCL) includes a small subset of the entire class library and is the core set of classes that serve as the basic API of the Common Language Runtime.^[9] The classes in mscorlib.dll and some of the classes in System.dll and System.core.dll are considered to be a part of the BCL. The BCL classes are available in both .NET Framework as well as its alternative implementations including .NET Compact Framework, Microsoft Silverlight and Mono.

The Framework Class Library (FCL) is a superset of the BCL classes and refers to the entire class library those ships with .NET Framework. It includes an expanded set of libraries, including Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation among others. The FCL is much larger in scope than standard libraries for languages like C++, and comparable in scope to the standard libraries of Java.

Memory management

The .NET Framework CLR frees the developer from the burden of managing memory (allocating and freeing up when done); it handles memory management itself by detecting when memory can be safely freed. Memory is allocated instantiations of .NET types (objects) from the managed heap, a pool of memory managed by the CLR. As long as there exists a reference to an object, which might be either a direct reference to an object or via a graph of objects, the object is considered to be in use. When there is no reference to an object, and it cannot be reached or used, it becomes garbage, eligible for collection. .NET Framework includes a garbage collector which runs periodically, on a separate thread from the application's thread, that enumerates all the unusable objects and reclaims the memory allocated to them.

The .NET Garbage Collector (GC) is a non-deterministic, compacting, mark-and-sweep garbage collector. The GC runs only when a certain amount of memory has been used or there is

enough pressure for memory on the system. Since it is not guaranteed when the conditions to reclaim memory are reached, the GC runs are non-deterministic. Each .NET application has a set of roots, which are pointers to objects on the managed heap (managed objects). These include references to static objects and objects defined as local variables or method parameters currently in scope, as well as objects referred to by CPU registers.^[10] When the GC runs, it pauses the application, and for each object referred to in the root, it recursively enumerates all the objects reachable from the root objects and marks them as reachable. It uses CLI metadata and reflection to discover the objects encapsulated by an object, and then recursively walk them. It then enumerates all the objects on the heap (which were initially allocated contiguously) using reflection. All objects not marked as reachable are garbage. This is the mark phase. Since the memory held by garbage is not of any consequence, it is considered free space. However, this leaves chunks of free space between objects which were initially contiguous. The objects are then compacted together to make used memory contiguous again. Any reference to an object invalidated by moving the object is updated by the GC to reflect the new location. The application is resumed after the garbage collection is over.

The GC used by .NET Framework is actually generational.¹ Objects are assigned a generation; newly created objects belong to Generation 0. The objects that survive a garbage collection are tagged as Generation 1, and the Generation 1 objects that survive another collection are Generation 2 objects. The .NET Framework uses up to Generation 2 objects. Higher generation objects are garbage collected less frequently than lower generation objects. This helps increase the efficiency of garbage collection, as older objects tend to have a longer lifetime than newer objects. Thus, by removing older (and thus more likely to survive a collection) objects from the scope of a collection run, fewer objects need to be checked and compacted.

SQL SERVER:

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internetfacing applications with many concurrent users.

SQL Server includes better compression features, which also helps in improving scalability.^[21] It enhanced the indexing algorithms and introduced the notion of filtered indexes. It also includes Resource Governor that allows reserving resources for certain users or workflows. It also includes capabilities for transparent encryption of data (TDE) as well as compression of backups. SQL Server 2008 supports the ADO.NET Entity Framework and the reporting tools, replication, and data definition will be built around the Entity Data Model.^[22] SQL Server Reporting Services will gain charting capabilities from the integration of the data visualization products from Dundas Data Visualization, Inc., which was acquired by Microsoft.^[23] On the management side, SQL Server 2008 includes the Declarative Management Framework which allows configuring policies and constraints, on the entire database or certain tables, declaratively.

The version of SQL Server Management Studio included with SQL Server 2008 supports IntelliSense for SQL queries against a SQL Server 2008 Database Engine. SQL Server 2008 also makes the databases available via Windows PowerShell providers and management functionality

5. PROJECT DESCRIPTION

MODULES

ADMIN

- Login
- Add Students
- View Students
- View Attendance

USER

- Login
- Fingerprint Verification □ Attendance

5.1 MODULES DESCRIPTION

ADMIN:

Login

The Admin login feature provides secure access to the administrative interface. Admins can log in using unique credentials to gain privileged access to the system's administrative functions.

Add Students

This functionality allows administrators to add new students to the system. Admins can input necessary student details, including identification information, which will be associated with their respective fingerprints for attendance tracking.

View Students

The View Students feature enables administrators to access a comprehensive list of all registered students. This includes essential information about each student, aiding in efficient management and oversight.

View Attendance:

Admins can utilize the View Attendance feature to monitor and analyze attendance records. This includes viewing attendance data for individual students or generating reports to assess attendance trends across various timeframes.

USER

Login

The User login feature enables students to securely access their individual accounts within the system. Students use their unique login credentials to gain access to the system.

Fingerprint Verification

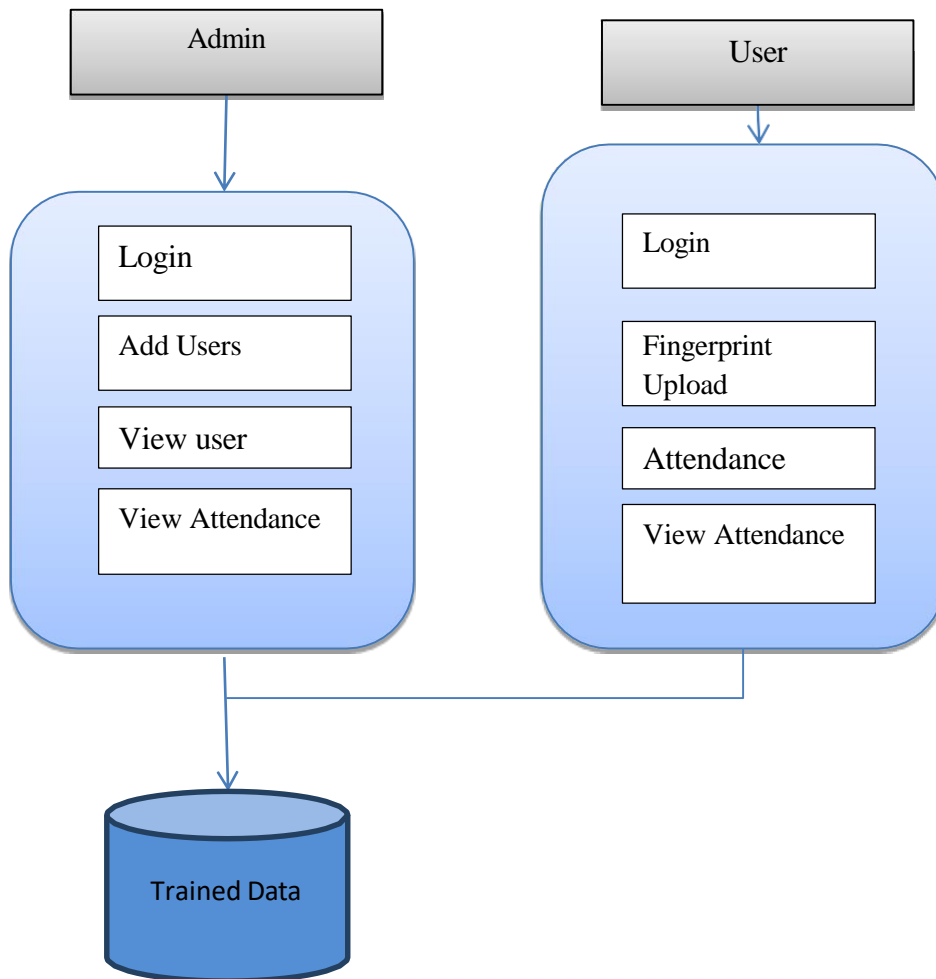
Students undergo fingerprint verification to confirm their identity securely. This biometric authentication process ensures that the attendance record is accurate and linked to the correct student.

Attendance

Upon successful fingerprint verification, students can mark their attendance. The system records the timestamp and associates it with the student's profile, providing real-time and accurate attendance tracking.

6.SYSTEM DESIGN AND IMPLEMENTATION

ARCHITECTURE DIAGRAM

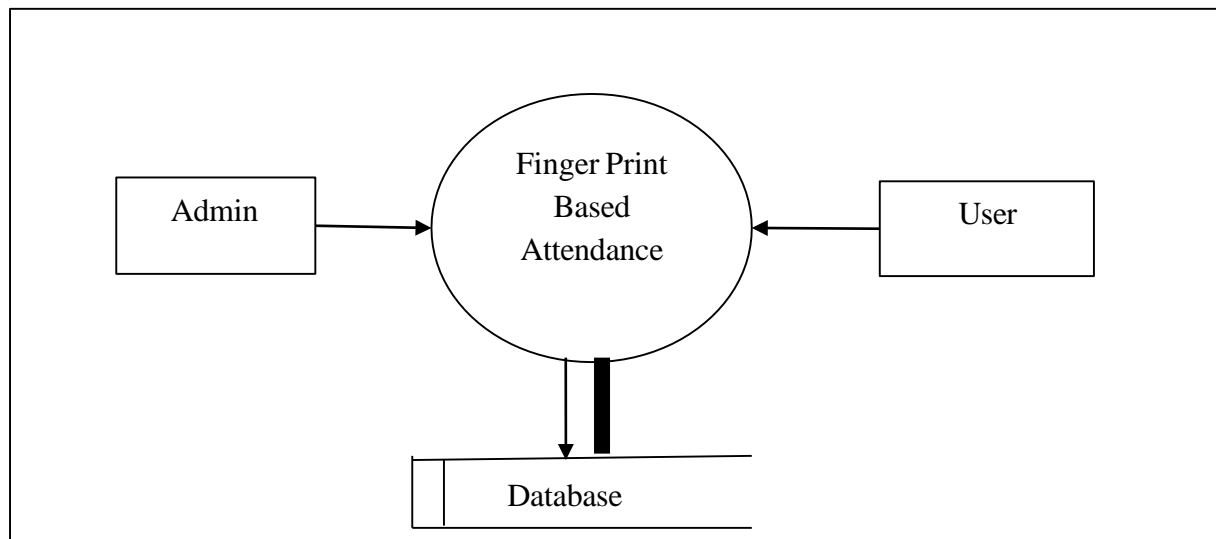


DATAFLOW DIAGRAM

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

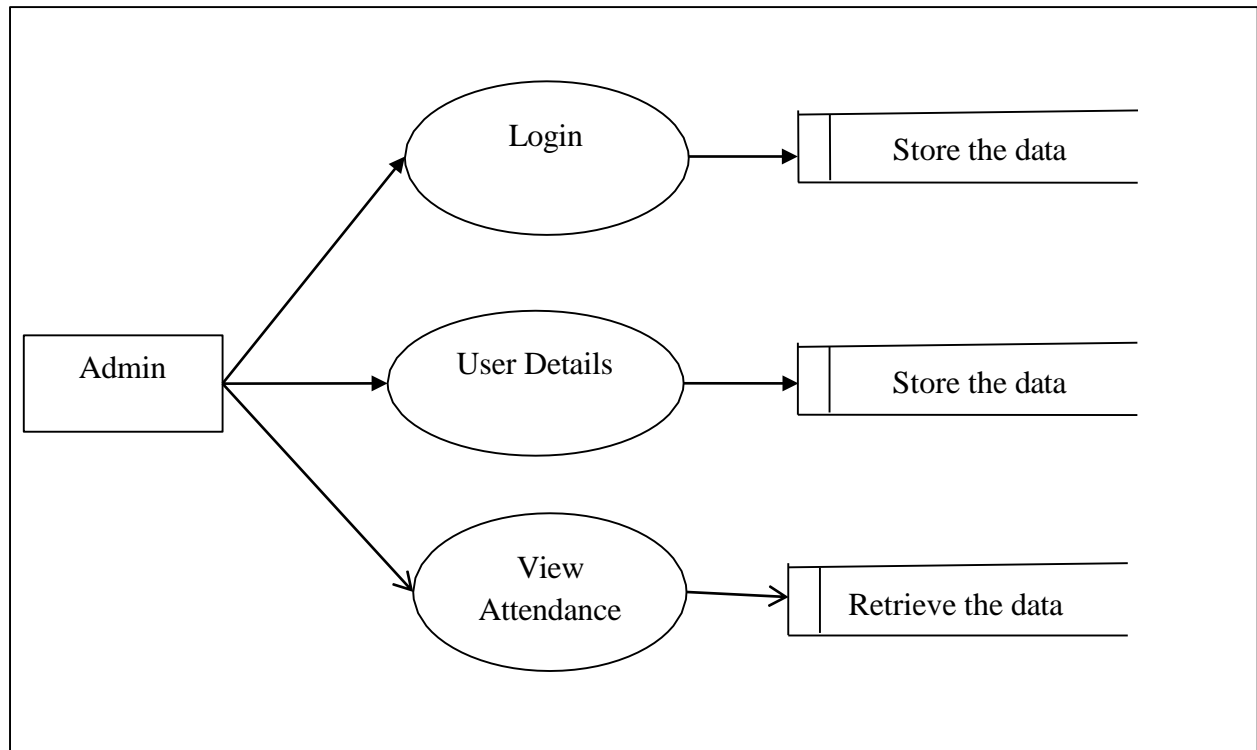
The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present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



Level 0 Data flow diagram

LEVEL-1

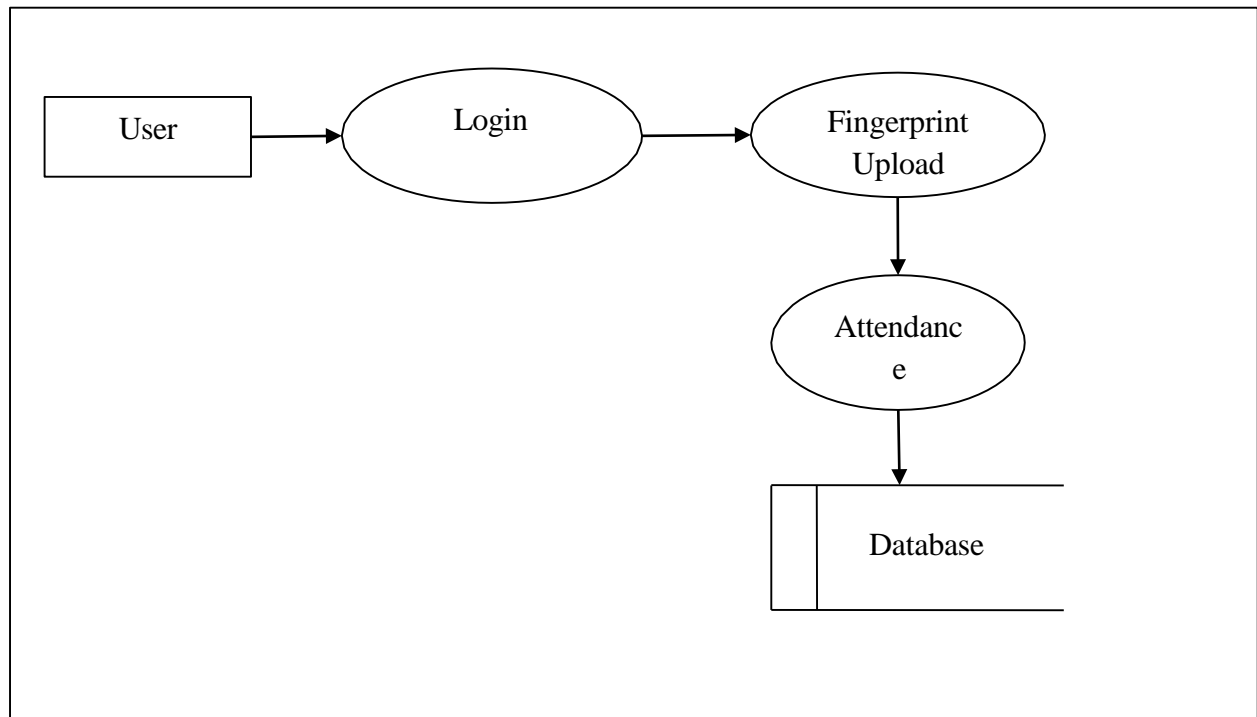
The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable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



Level 1 Data flow diagram

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 modeling and one of the oldest



Level 2 Data flow diagram

7. SYSTEM TESTING:

A test case is an asset of data that the system will process as normal input. The strategies that we have used in our project are, **System Testing:**

Testing is the stage of implementation of which aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct the goal will be achieved. The candidates system subject to a variety of tests. Online response, volume, stress, recovery, security and usability tests. A series of testing are performed for the proposed system before the system is ready for user acceptance testing.

Unit Testing:

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated .Then the web form level is made.

Integration Testing:

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus the system testing is a confirmation that all its correct and an opportunity to show the user that the system works.

Validation Testing:

The final step involves validation testing which determines whether the software function as the user expected. The end-user rather than the system developer conduct this test most software developers as a process called “Alpha and Beta test” to uncover that only the end user seems able to find. The compilation of the entire project is based on the full satisfaction of the end users.

Acceptance Testing:

- Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages.
- The function of performance characteristics to specification and is accepted.
- A deviation from specification is uncovered and a deficiency list is created.

White Box Testing

White box testing, sometimes called "Glass-box testing". Using white box testing methods, the following tests were made on the system,

- All independent paths with in a module have been exercised at least once.
- All logical decisions were checked for the true and false side of the values.
- All loops were executed to check their boundary values.
- Internal data-structure was tested for their validity.

Black Box Testing:

Black box testing focuses on the functional requirements of the software. That is black box testing enables the software engineer to drive a set of input conditions that will fully exercise the requirements for a program. Black box testing is not an alternative for white box testing techniques. Rather, it is a complementary approach that is likely to uncover different class of errors. Black box testing attempts to find errors in the following categories:

- Interfingerprint errors.
- Performances in data structures or external database access.
- Performance errors.
- Initialization and termination errors.
- Incorrect or missing functions.

8. CONCLUSION

Capturing the images from camera or cc camera and applying techniques fingerprint detection and recognition can decrease the manual work from human and increase the security safety, taking the decision from this recognition result. Based on this fingerprint detection and recognition can used in implement so many application like automatic attendances system based on fingerprint recognition, worker attendances, security, safety, police application like finding thief in image that help to catching thief. In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant an record student's attendance. It saves time and effort, especially if it is a lecture with huge number of students. The complete system is implemented in Python. This attendance system shows the use of facial recognition techniques for the purpose of student attendance and for the further process this record of student can be used in exam related issues.

9. FEATURE ENHANCEMENT

In future work we can implement this concept in palm print authentication system with template protection using topology code for local registration and security enhancement.

10. APPENDEX

10.1 SOURCE CODE

CODING:

Home

```
<% @ Page Title="" Language="C#" MasterPageFile="~/Login.master"
AutoEventWireup="true" CodeFile="Addpublic.aspx.cs" Inherits="Addpublic" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

    <form id="frm" runat="server">

        <div class="container-fluid" >

            <div class="row">

                <div class="row">

                    <div class="col-sm-2" >

                        </div>

                        <div class="col-sm-2" >

                            </div>

                            <div class="col-sm-1" >

                                </div>

                                <div class="col" >
```

```
<p style="font-family :18thCentury;font-size :x-large;font-weight :bold">ADD STUDENT  
DETAILS</p>
```

```
</div>
```

```
</div>  
<br />
```

```
<div class="col-sm-12 bg" >
```

```
<br />
```

```
<div class="row">
```

```
<div class="col-sm-2" >
```

```
<p>Name</p>
```

```
</div>
```

```
<div class="col-sm-3" >
```

```
<asp:TextBox ID="txtname" Width ="200" TabIndex="1" runat="server" > </asp:TextBox>
```

```
</div>
```

```
<div class="col-sm-2" >
```

```
<p>Year</p>
```

```
</div>
```

```
<div class="col-sm-3" >
```

```
<asp:TextBox ID="txtcountry" Width ="200" TabIndex ="7" runat="server" > </asp:TextBox>
```


</div>

</div>

<div class ="row">

<div class ="col-sm-2" >

<p>Gender</p>
</div>

<div class ="col-sm-3" >

<asp:RadioButtonList ID="rbgender" CellPadding="5" CellSpacing ="5" TabIndex ="2"
runat="server" RepeatDirection="Horizontal">

<asp:ListItem Text="Male" Value="Male"></asp:ListItem>

<asp:ListItem Text="Female" Value="Female"></asp:ListItem>

</asp:RadioButtonList>

</div>

<div class ="col-sm-2" >

<p>Department</p>

</div>

<div class ="col-sm-3" >

<asp:TextBox ID="txtstate" Width ="200" TabIndex ="8" runat="server" > </asp:TextBox>

```

</div>

</div>

<br />

<div class="row">

<div class="col-sm-2" >

<p>DOB</p>

</div>

<div class="col-sm-3" >
<asp:TextBox ID="txtdob" Width ="200" TabIndex ="3" runat="server" >
</asp:TextBox>

</div>

<div class="col-sm-2" >

<p>City</p>

</div>

<div class="col-sm-3" >

<asp:TextBox ID="txtcity" Width ="200" TabIndex ="9" runat="server" >
</asp:TextBox>

</div>

</div>

<br />

```

```
<div class ="row">
```

```
<div class ="col-sm-2" >
```

```
<p>Email</p>
```

```
</div>
```

```
<div class ="col-sm-3" >
```

```
<asp:TextBox ID="txtemail" Width ="200" TabIndex ="4" runat="server" >  
</asp:TextBox>
```

```
</div>
```

```
<div class ="col-sm-2" >
```

```
<p>Pin Code</p>
```

```
</div>
```

```
<div class ="col-sm-3" >
```

```
<asp:TextBox ID="txtpincode" Width ="200" TabIndex ="10" runat="server" >  
</asp:TextBox>
```

```
</div>
```

```
</div>
```

```
<br />
```

```
<div class ="row">
```

```
<div class ="col-sm-2" >
```

```
<p>Contact No</p>
```

```
</div>
```

```
<div class="col-sm-3" >
  <asp:TextBox ID="txtcontact" Width="200" TabIndex="5" runat="server" >
</asp:TextBox>
```

```
</div>
```

```
<div class="col-sm-2" >
```

```
<p>Address</p>
```

```
</div>
```

```
<div class="col-sm-3" >
```

```
<asp:TextBox ID="txtaddress" TextMode="MultiLine" TabIndex="11" Width
="200" runat="server" > </asp:TextBox>
```

```
</div>
```

```
</div>
```

```
<br />
```

```
<br />
```

```
<div class="row">
```

```
<div class="col-sm-2" >
```

```
</div>
```

```
<div class="col-sm-3" >
```

```
</div>
```

```
<div class="col-sm-2" >
```

```
<p>Finger</p>
```

</div>

<div class ="col-sm-3" >

<asp:FileUpload ID="FileUpload1" runat="server" />

</div>

</div>

<div class ="row">

<div class ="col-sm-2" >

<p>User Name</p>

</div>

<div class ="col-sm-3" >

<asp:TextBox ID="txtauthornumber" TextMode ="SingleLine" TabIndex ="11"
Width ="200" runat="server" > </asp:TextBox>

</div>

<div class ="col-sm-2" >

<p>Password</p>

</div>

<div class ="col-sm-3" >

```

        <asp:TextBox ID="txtpassword" TextMode="Password" TabIndex="12"
Width="200" runat="server" > </asp:TextBox>

        </div>

        </div>

        <br />

        <br />
        <div class="row">

            <div class="col-sm-2" >

                <button id="btnnewuser" type="button" runat="server" tabindex="13" class="btn
btnprimary">Clear</button>

                </div>

                <div class="col-sm-10" >

                    <button id="btnsubmit" type="button" onclick="" runat="server" class="btn
btnprimary">

                        <asp:Button ID="btnsubmit1" runat="server" BackColor
="Transparent" TabIndex="14" BorderWidth="0px" OnClick="btnsubmit_Click"
Text="Submit" />

                    </button>

                </div>

            </div>

        <br />

```


</div>

</div>

</form>

```
</asp:Content> using System;  
  
using System.Collections.Generic;  
  
using System.Linq; using  
  
System.Web; using  
  
System.Web.UI; using  
  
System.Web.UI.WebControls;  
  
using System.Data.SqlClient;
```

```

using System.Data; using
System.IO; public partial class
Addpublic : System.Web.UI.Page

{

    SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\hts\Desktop\MK\2022-
23\Projects\finger_Print_attendance\Attendance\App_Data\Attendance.mdf;Integrated
Security=True");    SqlCommand cmd;    protected void Page_Load(object sender,
EventArgs e)

    {

    }

    protected void btnsubmit_Click(object sender, EventArgs e)

    {

        string filePath, fileName;
        if (FileUpload1.PostedFile != null)

        {
            filePath = FileUpload1.PostedFile.FileName; // file name
with path.        fileName = FileUpload1.FileName; // Only file name.

        }

```



```

        string filename = Path.GetFileName(FileUpload1.PostedFile.FileName);

string ct = Path.GetFileName(FileUpload1.PostedFile.ContentType);


FileUpload1.SaveAs(Server.MapPath("FingerUpload/" + filename));


System.Drawing.Image img =
System.Drawing.Image.FromStream(FileUpload1.PostedFile.InputStream);

decimal height = img.Height;        decimal width = img.Width;


        decimal size = Math.Round((((decimal)FileUpload1.PostedFile.ContentLength /
(decimal)1024),                2);

con.Open();


        cmd = new SqlCommand("insert into publictbl values
( @Name,@gender,@dob,@emailid,@contactno,@country,@state,@city,@pincode,@address,
@aadhar,@pwd,@path,@filename,@hpix,@wpix,@fsize)",con);

cmd.Parameters.AddWithValue("@Name",txtname .Text );

cmd.Parameters.AddWithValue("@gender",rbgender.Text);

cmd.Parameters.AddWithValue("@dob", txtdob.Text);

cmd.Parameters.AddWithValue("@emailid", txtemail.Text);

cmd.Parameters.AddWithValue("@contactno", txtcontact.Text);

cmd.Parameters.AddWithValue("@country", txtcountry.Text);

cmd.Parameters.AddWithValue("@state", txtstate.Text);

cmd.Parameters.AddWithValue("@city", txtcity.Text);

cmd.Parameters.AddWithValue("@pincode", txtpincode.Text);

```

```

cmd.Parameters.AddWithValue("@address", txtaddress .Text);

cmd.Parameters.AddWithValue("@aadhar", txtauthornumber .Text);

cmd.Parameters.AddWithValue("@pwd", txtpassword.Text);

cmd.Parameters.AddWithValue("@path", "FingerUpload/" + filename);

cmd.Parameters.AddWithValue("@filename", filename);

cmd.Parameters.AddWithValue("@hpix", height);

cmd.Parameters.AddWithValue("@wpix", width);

cmd.Parameters.AddWithValue("@fsize", size);      cmd.ExecuteNonQuery();

con.Close();

```

```

    Response.Write("<script>alert ('Record Saved. ....')</script>");

```

```

    }      private void

clear()

    {      txtname.Text =

"";      txtaddress.Text =

""; txtcity.Text = "";

txtcontact.Text = "";

txtcountry.Text = "";

txtdob.Text = "";

txtemail.Text = "";

txtauthornumber.Text =

```

```

"";    txtpassword.Text
= "";    txtpincode.Text
= "";    txtstate.Text =
"";

    }

}

```

Admin Login

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Login.master"
AutoEventWireup="true" CodeFile="AdminLogin.aspx.cs" Inherits="AdminLogin" %>

<asp:Content ID="Content1" runat="server" contentplaceholderid="ContentPlaceholder1">

    <br />

    <br />

    <p>

        <form id = "frm" runat = "server">

            <div class = "container-fluid" >

                <div class="row">
                    <div class="row">

                        <div class = "col-sm-2" >

                            </div>

```

```
<div class ="col-sm-2" >
```

```
</div>
```

```
<div class ="col-sm-1" >
```

```
</div>
```

```
<div class ="col-lg-6" >
```

```
    <p style ="font-family :18thCentury;font-size :x-large;font-weight :bold">Admin  
Login</p>
```

```
</div>
```

```
</div>
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<div class="col-sm-12 bg" >
```

```
<div class ="row">
```

```
    <div class ="col-sm-2" >
```

```
        <p>User Name</p>
```

</div>

<div class ="col-sm-10" >

<asp:TextBox ID="txtuname" Width ="200" runat="server" > </asp:TextBox>

</div>

</div>

<div class ="row">

<div class ="col-sm-2" >

<p>Password</p>

</div>

<div class ="col-sm-10" >

<asp:TextBox ID="txtpwd" TextMode ="Password" Width ="200" runat="server" >
</asp:TextBox>

</div>

</div>

<div class ="row">

<div class ="col-sm-2" >

<button id="btnnewuser" type="button" runat="server" class="btn btn-primary">

<asp:Button ID="Button2" BackColor="Transparent" OnClick="Button2_Click"
BorderWidth="0px" Text="Cancel" runat="server" />
</button>

</div>

<div class="col-sm-10">

<button id="btnsubmit" type="button" runat="server" class="btn btn-primary">

<asp:Button ID="Button1" BackColor="Transparent" OnClick="Button1_Click"
BorderWidth="0px" Text="Login" runat="server" />

</button>

</div>

</div>

</div>

</div>
</form>

</p>

```
</asp:Content> using System; using
System.Collections.Generic; using System.Linq; using
System.Web; using System.Web.UI; using
System.Web.UI.WebControls; public partial class
AdminLogin : System.Web.UI.Page

{    protected void Page_Load(object sender, EventArgs
e)

{

}

protected void Button1_Click(object sender, EventArgs e)

{
```

```

        if (txtuname.Text == "admin" && txtpwd.Text == "admin")

        {

            Response.Redirect("Adminhome.aspx");

        }
        else

        {

            Response.Write("<script>alert ('Username Password Mismatch')</script>");

        }

    }

    protected void Button2_Click(object sender, EventArgs e)

    {

    }

}

User Login

<% @ Page Title="" Language="C#" MasterPageFile="~/Login.master"
AutoEventWireup="true" CodeFile="Publiclogin.aspx.cs" Inherits="Publiclogin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

    <form id = "frm" runat = "server">

```



```
<div class ="container-fluid" >
```

```
<div class="row">
```

```
<div class ="col-sm-2" >
```

```
</div>
```

```
<div class ="col-sm-2" >
```

```
</div>
```

```
<div class ="col-sm-1" >
```

```
</div>
```

```
<div class ="col-sm-6" >
```

```
<p style ="font-family :18thCentury;font-size :x-large;font-weight :bold">User  
Login</p>
```

```
</div>
```

```
</div>
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<br />
```

```
<br />
```

```

<div class="col-sm-12 bg" >

    <div class ="row">

        <div class ="col-sm-2" >

            <p>User Name</p>

        </div>

        <div class ="col-sm-10" >

            <asp:TextBox ID="txtuname" Width ="200" runat="server" > </asp:TextBox>

        </div>
    </div>

    <br />

    <br />

    <div class ="row">

        <div class ="col-sm-2" >

            <p>Password</p>

        </div>

        <div class ="col-sm-10" >

            <asp:TextBox ID="txtpwd" TextMode ="Password" Width ="200" runat="server" >
</asp:TextBox>

        </div>
    </div>

```

```

</div>

<br />

<br />

<div class ="row">

    <div class ="col-sm-2" >

        <button id="btnnewuser" type="button" runat ="server" class="btn btn-primary">

            <asp:Button ID="Button2" BackColor="Transparent" OnClick="Button2_Click"
BorderWidth ="0px" Text="Cancel" runat="server" />

        </button>

    </div>

    <div class ="col-sm-10" >

        <button id="btnsubmit" type="button" runat ="server" class="btn btn-primary">

            <asp:Button ID="Button1" BackColor="Transparent" OnClick ="Button1_Click"
BorderWidth ="0px" Text="Login" runat="server" />

        </button>

    </div>

</div>

<br />

<br />

```

</div>

</div>

</form>

</asp:Content> using System; using

System.Collections.Generic; using System.Linq; using

System.Web; using System.Web.UI; using

System.Web.UI.WebControls; using

System.Data.SqlClient; using System.Data; public

partial class Publiclogin : System.Web.UI.Page {

protected void Page_Load(object sender, EventArgs e)

{

}

protected void Button1_Click(object sender, EventArgs e)

{

 SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\hts\Desktop\MK\2022-
23\Projects\finger_Print_attendance\Attendance\App_Data\Attendance.mdf;Integrated
Security=True"); SqlCommand cmd; con.Open();

```
cmd = new SqlCommand("select * from publictbl where aadhar='" + txtuname.Text + "' and  
pwd='" + txtpwd.Text + "'", con);
```

```
SqlDataReader dr = cmd.ExecuteReader();
```

```
if (dr.Read())
```

```
{
```

```
Session["uname"] = dr["aadhar"].ToString();
```

```
Session["mobile"] = dr["contactno"].ToString();
```

```
Session["fpath"] = dr["path"].ToString();  
Session["height"] = dr["hpix"].ToString();
```

```
Session["width"] = dr["wpix"].ToString();
```

```
Session["size"] = dr["fsize"].ToString();
```

```
Response.Redirect("FingerprintVerify.aspx");
```

```
}
```

```
else
```

```
{
```

```
Response.Write("<script>alert ('Username Password Mismatch')</script>");
```

```
}
```

```
}
```

```

protected void Button2_Click(object sender, EventArgs e)

{

    txtuname.Text      =      "";

txtpwd.Text = "";

}

}

```

Fingerprint Verification using

System; using

System.Collections.Generic; using

System.Linq; using System.Web;

using System.Web.UI; using

System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data; using

System.IO; public partial class

FingerprintVerify :

System.Web.UI.Page

```

{    public static string uheight,uwidth, useize, upath;

protected void Page_Load(object sender, EventArgs e)

{

    uheight =Session ["height"].ToString ();

uwidth    =    Session["width"].ToString();

useize = Session["size"].ToString();    upath

= Session["fpath"].ToString();


    Image1.ImageUrl = upath;

}

protected void Button2_Click(object sender, EventArgs e)

{

}

protected void Button1_Click(object sender, EventArgs e)

{    string filePath, fileName;

if (FileUpload1.PostedFile != null)

{

```

```

        filePath = FileUpload1.PostedFile.FileName; // file name with path.

fileName = FileUpload1.FileName;// Only file name.

    }

    string filename = Path.GetFileName(FileUpload1.PostedFile.FileName);

string ct = Path.GetFileName(FileUpload1.PostedFile.ContentType);

    FileUpload1.SaveAs(Server.MapPath("FingerVote/" + filename));

    System.Drawing.Image img =
System.Drawing.Image.FromStream(FileUpload1.PostedFile.InputStream);

decimal height = img.Height;    decimal width = img.Width;

    decimal size = Math.Round((((decimal)FileUpload1.PostedFile.ContentLength /
(decimal)1024), 0);    decimal oheight =

Convert.ToDecimal(uheight);    decimal owidth =

Convert.ToDecimal(uwidth);    decimal osize =

Convert.ToDecimal(usize);

    if ((height == oheight) && (width == owidth) && (size == osize))

    {

        Response.Redirect("Publichome.aspx");

    }

else

```



```

{

    Response.Write("<script>alert ('Finger Print Mismatch')</script>");

}

}

}

<% @ Page Title="" Language="C#" MasterPageFile="~/Login.master"
AutoEventWireup="true" CodeFile="FingerprintVerify.aspx.cs" Inherits="FingerprintVerify"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

    <form id = "frm" runat = "server">

<div class = "container-fluid" >

    <div class="row">

        <div class = "col-sm-2" >

            </div>

        <div class = "col-sm-2" >

            </div>
            <div class = "col-sm-1" >

                </div>

```

```

<div class ="col-sm-6" >

    <p style ="font-family :18thCentury;font-size :x-large;font-weight :bold">Public
Login</p>

</div>

</div>

<br />

<br />

<br />

<br />

<br />

<br />

<div class="col-sm-12 bg" >

    <div class ="row">

        <div class ="col-sm-2" >

            <p>User Name</p>

        </div>

        <div class ="col-sm-10" >

            <asp:TextBox ID="txtuname" Width ="200" runat="server" > </asp:TextBox>

```

</div>
</div>

<div class ="row">

<div class ="col-sm-2" >

<p>Finger Print</p>

</div>

<div class ="col-sm-10" >

<asp:Image ID="Image1" Width="100px" Height ="100px" runat="server" />

</div>

</div>


```

<div class ="row">

<div class ="col-sm-2" >

<p>Upload Finger Print</p>

</div>
<div class ="col-sm-10" >

<asp:FileUpload ID="FileUpload1" runat="server" />

</div>

</div>

<br />

<br />

<div class ="row">

<div class ="col-sm-2" >

<button id="btnnewuser" type="button" runat ="server" class="btn btn-primary">

<asp:Button ID="Button2" BackColor="Transparent" OnClick="Button2_Click"
BorderWidth ="0px" Text="Cancel" runat="server" />

</button>

</div>

<div class ="col-sm-10" >

<button id="btnsubmit" type="button" runat ="server" class="btn btn-primary">

```

```
<asp:Button ID="Button1" BackColor="Transparent" OnClick="Button1_Click"
BorderWidth ="0px" Text="Login" runat="server" />
```

```
</button>
```

```
</div>
```

```
</div>
```

```
<br />
```

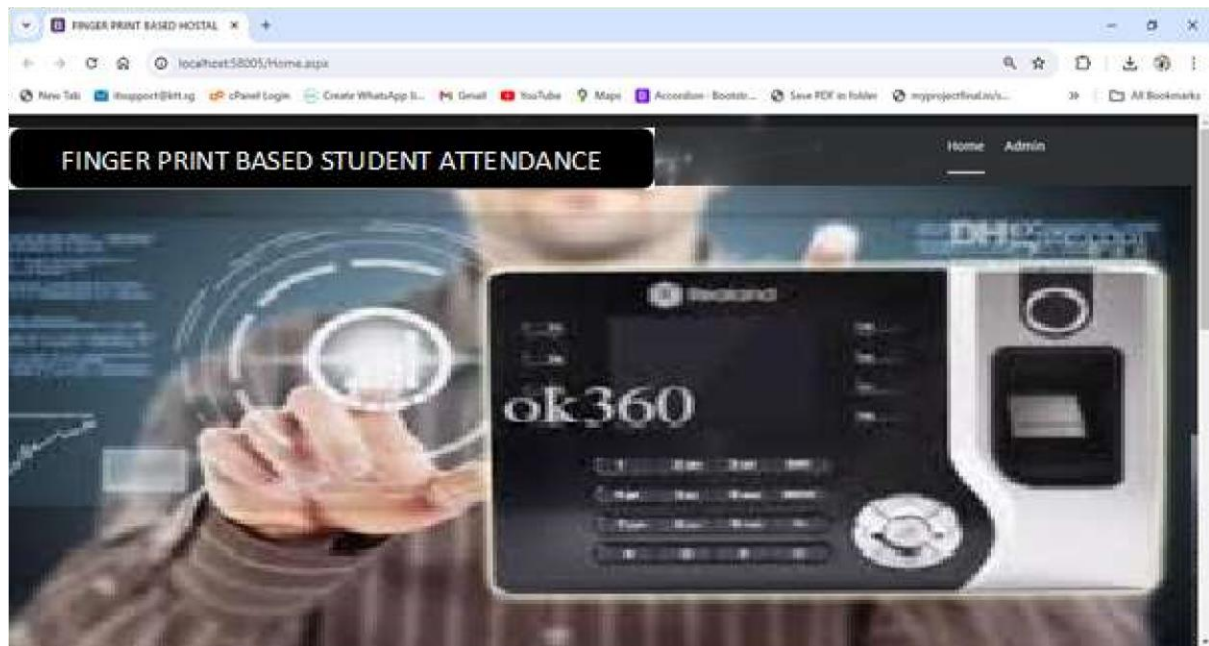
```
<br />
</div>
```

```
</div>
```

```
</form>
```

```
</asp:Content>
```

10.2 SCREEN SHOT



Login

FINGER PRINT BASED STUDENT ATTENDANCE

Home Admin

Admin Login

User Name:

Password:

Cancel Login

FINGER PRINT BASED STUDENT ATTENDANCE

Home Admin

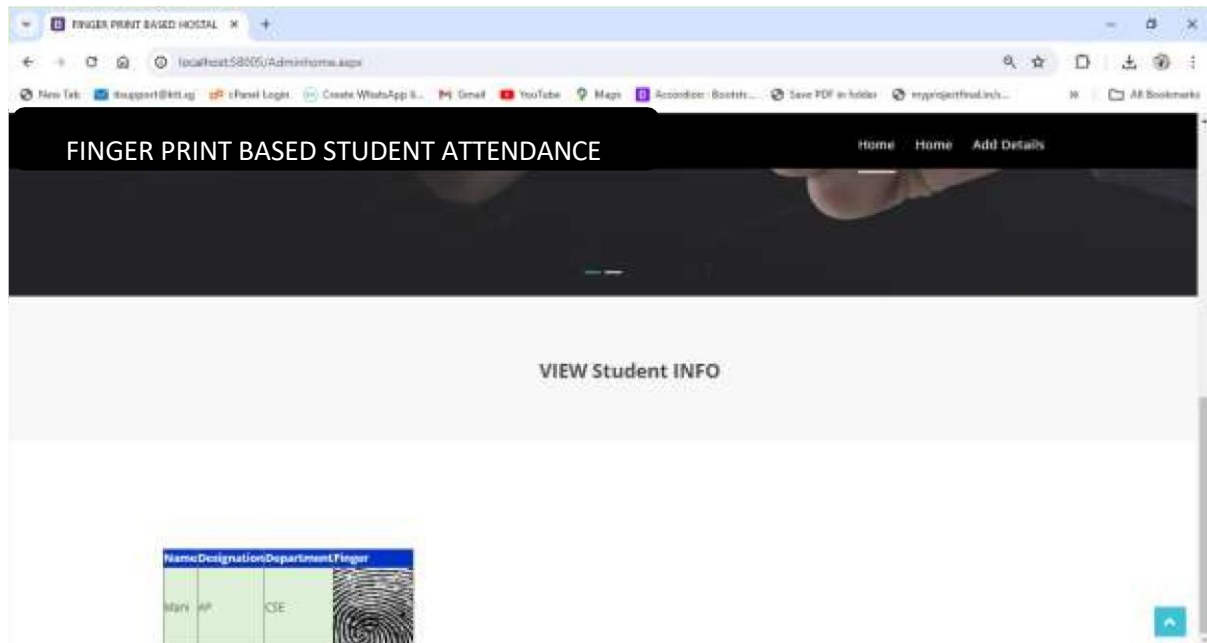
Admin Login

User Name:

Password:

Cancel Login

View Student



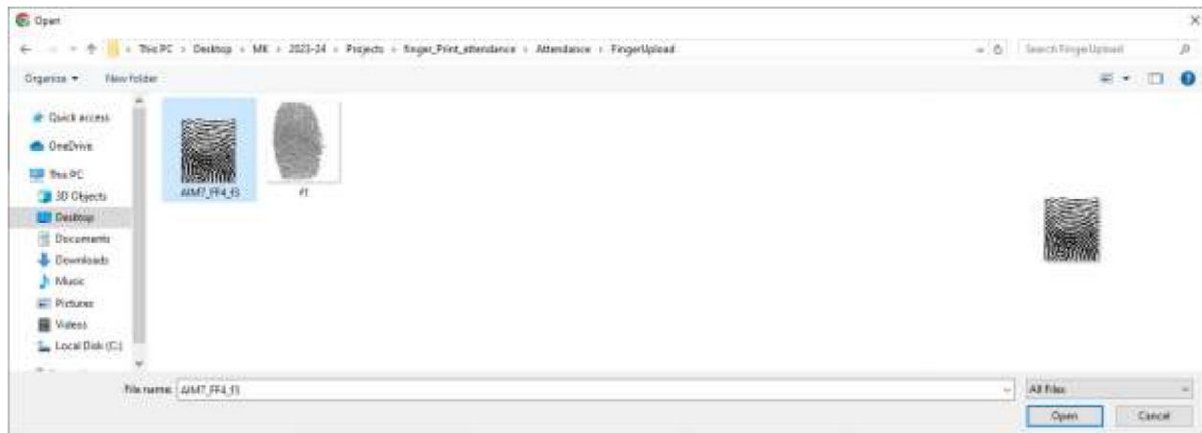
Add Student

The screenshot shows a web browser window with the title 'FINGER PRINT BASED STUDENT ATTENDANCE'. The address bar shows 'localhost:58005/Addpublic.aspx'. The page has a navigation bar with 'Home' and 'Admin' links. The main content area is titled 'ADD STUDENT DETAILS' and contains a form with the following fields:

Field	Value
Name	
Gender	<input type="radio"/> Male <input type="radio"/> Female
DOB	
Email	
Contact No	
Year	
Department	
City	
Pin Code	
Address	
Finger	<input type="button" value="Choose File"/> No file chosen
User Name	
Password	

At the bottom of the form are two buttons: 'Clear' and 'Submit'.

Upload Finger

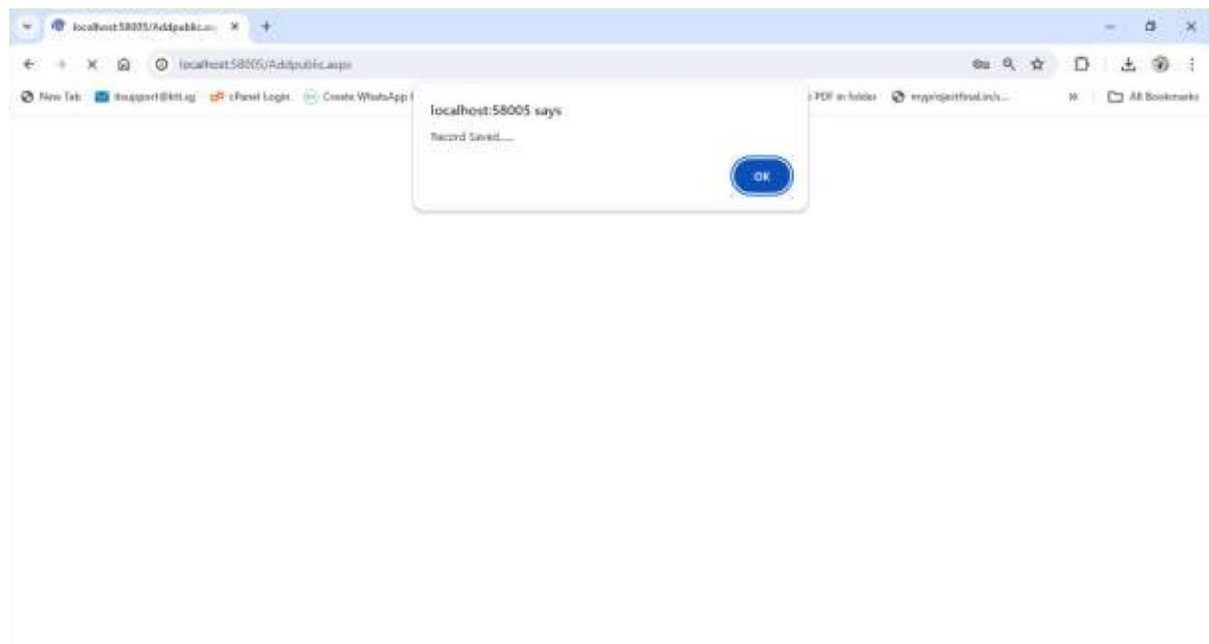


FINGER PRINT BASED HOSTAL

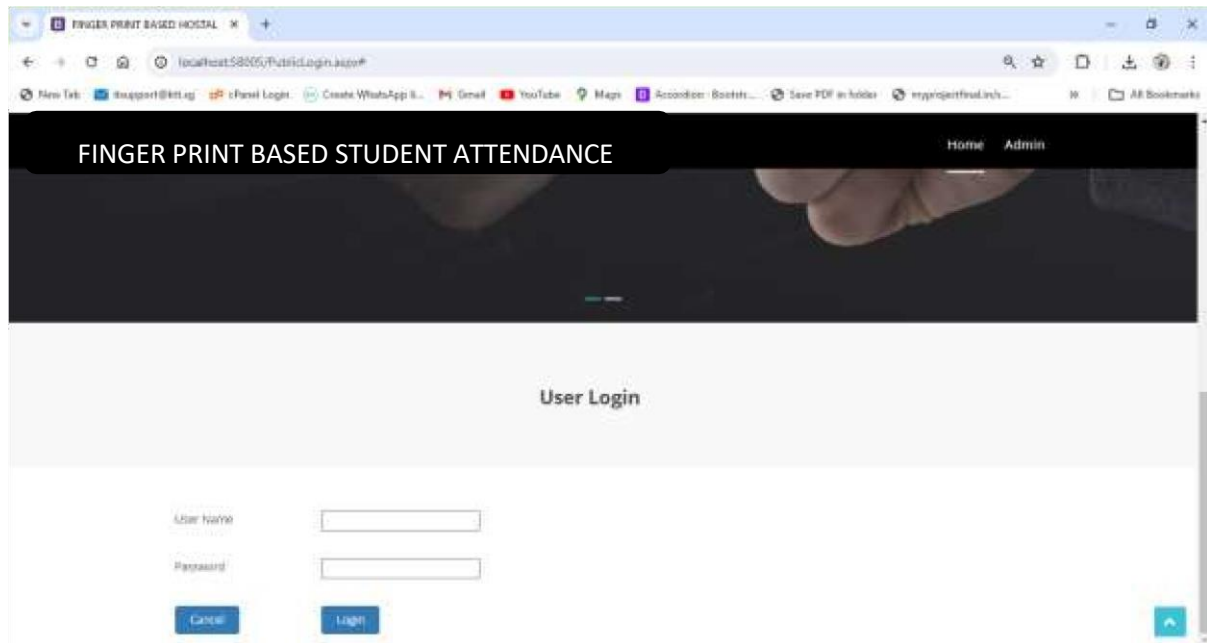
localhost:58005/Addpublic.aspx

ADD STUDENT DETAILS

Name	Mankandan Sacharam	Year	India
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female	Department	Tamilnadu
DOB		City	Trichy
Email	mankandan0994@gmail.com	Pin Code	620007
Contact No	7871361947	Address	Trichy
		Finger	Choose File AIM7_FT4_93.jpg
User Name	man	Password	



Login



The screenshot shows a web browser window with the title "FINGER PRINT BASED HOSTAL". The address bar shows the URL "localhost:58005/Html/Login.aspx". The browser's bookmark bar contains several items: "New Tab", "support@kktug", "iPanel Login", "Create WhatsApp B...", "Gmail", "YouTube", "Maps", "Accordion: Bootstrap...", "Save PDF in folder", "myprojectfinal.in...", and "All Bookmarks". The page has a dark header with the text "FINGER PRINT BASED STUDENT ATTENDANCE" and navigation links "Home" and "Admin". Below the header is a large image of a hand being scanned. The main content area is titled "User Login" and contains a login form with two input fields: "User Name" and "Password". Below these fields are two buttons: "Cancel" and "Login".

FINGER PRINT BASED STUDENT ATTENDANCE

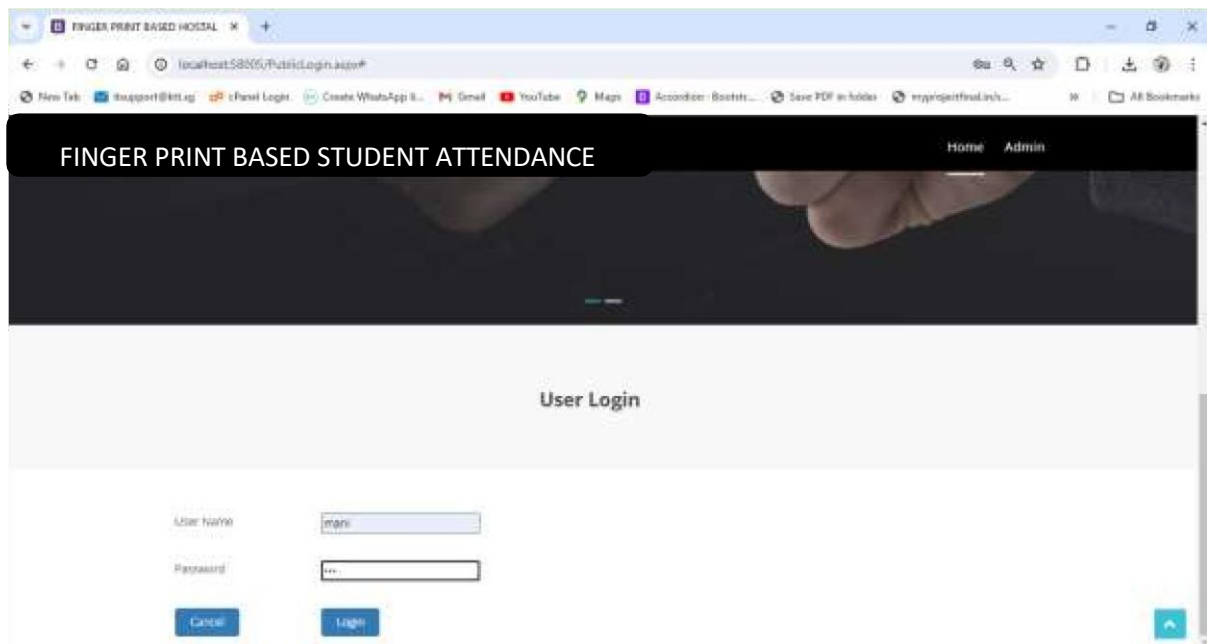
Home Admin

User Login

User Name

Password

Cancel Login



This screenshot is identical to the one above, but with the sample data "mani" entered in the "User Name" field and "123" entered in the "Password" field.

FINGER PRINT BASED STUDENT ATTENDANCE

Home Admin

User Login

User Name

Password

Cancel Login

Finger print Verify

The screenshot shows a web browser window with the title 'FINGER PRINT BASED HOSITAL'. The address bar shows 'localhost:58005/FingerPrintVerify.aspx'. The page has a header with a large 'E' logo. The main content area is titled 'Public Login'. Below the title, there is a form with three input fields: 'User Name' (a text box), 'Finger Print' (a box containing a fingerprint icon), and 'Upload Finger Print' (a box with a 'Choose File' button and the text 'No file chosen'). At the bottom of the form are two buttons: 'Cancel' and 'Login'.

The screenshot shows a web browser window with the title 'FINGER PRINT BASED STUDENT ATTENDANCE'. The address bar shows 'localhost:58005/FingerPrintVerify.aspx'. The page has a dark header with the title 'FINGER PRINT BASED STUDENT ATTENDANCE' and navigation links 'Home' and 'Admin'. The main content area is titled 'Public Login'. Below the title, there is a form with three input fields: 'User Name' (a text box containing 'mani'), 'Finger Print' (a box containing a fingerprint icon), and 'Upload Finger Print' (a box with a 'Choose File' button and the text '14007_FFL15.jpg'). At the bottom of the form are two buttons: 'Cancel' and 'Login'.

11. REFERENCES:

BIBLIOGRAPHY

- Advanced .NET Remoting 2nd Edition (Ingo Rammer and Mario Szpuszta, Apress, March 2005)
- .NET to .NET Migration Handbook (Christian Nagel et al, Wrox, January 2003) ➤ OF .NET (Christian Nagel et al, Wrox, September 2001)
- Data-Centric .NET Programming (Christian Nagel et al, Wrox, December 2001)
- Professional .NET Network Programming 2nd Edition (Christian Nagel et al, Wrox, September 2004)
- Professional .NET (Christian Nagel et al, Wrox, March 2002)
- Professional .NET Web Services (Christian Nagel et al, Wrox, December 2001)
- The Joy of .NET Programming: A Beginner's Guide – by Alan Forbes
- .NET & MySQL Novice to Ninja – by Kevin Yank
- Head First .NET & MySQL – by Lynn Beighley & Michael Morrison
- .NET: A Beginner's Guide – by Vikram Vaswani
- Murach's .NET & MySQL – by Joel Murach & Ray Harris
- Learning .NET, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites – by Robin Nixon
- Programming .NET – by Kevin Tatroe, Peter MacIntyre & Rasmus Lerdorf “Foreword By: Michael Bourque”
- .NET & MySQL Web Development – by Luke Welling & Laura Thompson