FACE RECOGNITION BASED ATTENDANCE

Project Report Submitted by

HANAN NESHRIN P (GGATBCA017)

MOHAMMED SHEKEEB K (GGATBCA011)

MOHAMMED SHINADH P (GGATBCA033)

SHAHMA CHETTALI (GGATBCA021)

Under the Guidance Of

Mrs. RAHMA P, MSc. Computer Science

(Asst. Professor – Department of Computer Science)

In Partial Fulfillment of the Requirements

BACHELOR OF COMPUTER APPLICATIONS

Of University of Calicut

DEPARTMENT OF BCA

GEMS ARTS AND SCIENCE COLLEGE

RAMAPURAM, MALAPPURAM – 679350

2019-22

A Unit of Global Education and Management Studies Co-Operative Ltd.No.M782
(Affiliated to University of Calicut)
P.O. Ramapuram, Malappuram Dt, Kerala Pin- 679 321,
Ph.: 0493-289162, 256123, 287100
www.gemseducation.org Email: gemsasc@gmail.com

CHAIRMAN	**PRINCIPAL**
MANJALAMKUZHI ALI	**Ph.:0493-3256122**

CERTIFICATE

This is to certify that **HANAN NESHRIN P (GGATBCA017), MOHAMMED SHEKEEB K (GGATBCA011), MOHAMMED SHINADH P (GGATBCA033), SHAHMA CHETTALI (GGATBCA021)** are the students of **BACHELOR OF COMPUTER APPLICATIONS, GEMS ARTS AND SCIENCE COLLEGE, RAMAPURAM** and the project report entitled on **"FACE RECOGNITION BASED ATTENDANCE"** has been prepared by them in partial fulfilment of the requirement for the award of BACHELOR OF COMPUTER APPLICATIONS degree of University of Calicut.

Place: Ramapuram

Head of Department:

Date:

GEMS ARTS AND SCIENCE COLLEGE RAMAPURAM

CERTIFICATE

This is to certify that this project work entitled **“FACE RECOGNITION BASED ATTENDANCE”** is submitted to **CALICUT UNIVERSITY** in partial fulfilment of the requirements for the award of the **Degree of Bachelor of Computer Applications** is record of the original work done by **HANAN NESHRIN P, MOHAMMED SHEKEEB K, MOHAMMED SHINADH P, SHAHMA CHETTALI** during the year 2021-2022 in the department of Computer Applications, GEMS College, Ramapuram, Malappuram.

Submitted for the university project viva-voice examination held on …./…./2022.

Internal Examiner: **External Examiner:**

CERTIFICATE

This is to certify that this project work entitled **“FACE RECOGNITION BASED ATTENDANCE”** is based on an original project study conducted and done by **HANAN NESHRIN P (GGATBCA017), MOHAMMED SHEKEEB K (GGATBCA011), MOHAMMED SHINADH P (GGATBCA033), SHAHMA CHETTALI (GGATBCA021)** under my supervision and guidance.

Place: Ramapuram **Faculty Guide**

Date:

DECLARATION

We **HANAN NESHRIN P, MOHAMMED SHEKEEB K, MOHAMMED SHINADH P, SHAHMA CHETTALI** hereby declare that the project entitled **"FACE RECOGNITION BASED ATTENDANCE"** submitted to Calicut University in partial fulfillment of the requirements for the award of degree of **BACHELOR OF COMPUTER APPLICATIONS**, is a record of original work done by during our period of our study at **GEMS ARTS AND SCIENCE COLLEGE, RAMAPURAM** under the supervision and guidance of **MRS. RAHMA P, MSc. Computer Science, Asst. Professor, Dept. of Computer Science.**

Place: Ramapuram **Signature of Candidate**

Date: **HANAN NESHRIN**

MOHAMMED SHEKEEB K

MOHAMMED SHINADH P

SHAHMA CHETTALI

ACKNOWLEDGEMENT

A project of this dimension would not have been possible without the genuine help and earnest support provided to us from all sources that we approached. This space would not be enough for us to mention all such persons here, but that would not be lessening our gratitude towards them. But it would be unfair if we do not express our sincere thanks to those persons who have provided us with the greatest support and guidance.

We wish to express our heartfelt gratitude to **Mr. Naveen Mohan, Principal, GEMS Arts and Science College, Ramapuram** for his encouragement and inspiring guidance throughout preparation of this project. We are very much thankful to our guide **Mrs. Rahma P,** Asst Professor, Department of Computer Science and **Mr. Hari,** Asst Professor and Head of Department of Bachelor of Computer Applications, GEMS Arts and Science College, Ramapuram for their valuable guidance for completing the project successfully. We take this opportunity to express our profound thoughts.

Last but not least, we thank our family and friends without who’s moral and emotional support, this project would not have materialized. They have been instrumental in keeping our confidence level high and hopes alive.

INDEX

INTRODUCTION

1.1ABSTRACT

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day to day activities of handling student attendance systems. Face recognition-based attendance system is a process of recognizing the students face for taking attendance by using face biometrics based on high - definition monitor video and other information technology. In my face recognition project, a computer system will be able to find and recognize human faces fast and precisely in images or videos that are being captured through a surveillance camera. Numerous algorithms and techniques have been developed for improving the performance of face recognition but the concept to be implemented here is Deep Learning. It helps in conversion of the frames of the video into images so that the face of the student can be easily recognized for their attendance so that the attendance database can be easily reflected automatically.

1.2 MODULE DESCRIPTION

Modules in the system are:

- □ Admin module
- □ Teacher module
- □ Student module
- □ Parent module

Admin

Under this module admin can register and modify department, teacher and student details.

- Login
- Student Registration
- Department Registration
- Teacher Registration

Teacher

Under this module teachers can add subjects and internal marks that are provided to each student for each subject. Teachers can view the attendance percentage of students and can edit attendance if needed.

- Login
- Subject Registration
- Add Internal Mark
- View Internal Mark
- View Attendance
- Performance Grade
- View Feedback
- Add Comments

Student

Under this module students can view their attendance percentage and internal marks provided to them for each subject. Students can also add their feedback.

- Login
- View Attendance
- View Internal Marks
- Add Feedback
- Performance Grade

Parent

Under this module parents can see their children's attendance percentage and the internal marks provided to them.

- Registration
- Login
- View Attendance
- View Internal Marks
- View Comments

SYSTEM ANALYSIS

2.1 SYSTEM SPECIFICATION

The following requirements are only the minimal requirements to run this utility more successfully and efficiently, there should be sufficient memory and software tools for efficient processing.

2.1.1 HARDWARE REQUIREMENTS

- Processor : 64 bit
- RAM : Min 3 GB
- Hard disk : 10 GB

2.1.2 SOFTWARE REQUIREMENTS

- Operating system : windows 7 or above, android
- Technology used : Python
- IDE :PyCharm , Android Studio
- Frame work : Flask
- Database : MYSQL

2.2 OVERVIEW OF PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of application

Python's features include –

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases** − Python provides interfaces to all major commercial database

- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable −** Python provides a better structure and support for large programs than shell scripting

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Features of Python Programming

- **A simple language which is easier to learn:** Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.
- **Free and open-source**: You can freely use and distribute Python, even for commercial use. Not only can you use and distribute softwares written in it, you can even make changes to Python's source code. Python has a large community constantly improving it in each iteration.
- **Portability**: You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.
- **Extensible and Embeddable**: Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.
- **A high-level, interpreted language:** Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations
- **Object-oriented**: Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating objects.
- **Large standard libraries to solve common tasks**: Python has a number of standard libraries which makes the life of a programmer much easier

since you don't have to write all the code yourself. For example: Need to connect a MySQL database on a Web server? You can use the MySQLdb library using import MySQL db . Standard libraries in Python are well tested and used by hundreds of people

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active. Let us write a simple Python program in a script. Python files have extension .py.

Python Identifiers

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus, **Manpower** and **manpower** are two different identifiers in Python.

Here are naming conventions for Python identifiers –

- Class names start with an uppercase letter. All other identifiers start with a lowercase letter.
- Starting an identifier with a single leading underscore indicates that the identifier is private.
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

Applications of Python

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.

However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use a language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework is normally a framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there is little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

Flask is an API of python that allows us to build web applications. It was developed by Armin Ronacher. Flask’s framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web application. A web application framework or web framework is the collection of modules and libraries that helps the developer to write applications without writing the low level codes such as protocols, thread management etc.

2.3. OVERVIEW OF MYSQL

Advantages of MYSQL

MYSQL is one of the top databases available in the market. MYSQL is a relational database with many advanced features and options. Over time , MYSQL has proved itself to be a fast , reliable and cost effective competitor to the other more expensive databases like MYSQL server and oracle . Here are a few advantages of using MYSQL in database development.

□ **Open source**

MYSQL is an open source database system which means that anyone can use it for free. Developers can amends its code to suit their requirements which means that MYSQL is highly customizable as well. Another edge of using MYSQL over other database systems is that it is available widely in the market with no ownership cost.

□ **Fast Development**

A lot of people around the globe are continuously developing new modules for integration with MYSQL. This means that it has a wider and faster development circle. Patches, upgrades and fixes are developed fast and become available in forums , blogs and developer sites on the internet.

□ **Better for Small Businesses**

This relational database system is free so it reduces the cost of overall database solution for small businesses and companies. This database is relatively easy to learn and operate , so operational cost is reduced substantially which is again an important factor in classifying MYSQL as an applicable and practical tool for small businesses.

□ **Cross Platform Operability**

MYSQL is easily installable and operable on different platforms including Windows , Linux , OS2 and Solaris . Cross platform operability makes it a favourable choice for development companies. The MYSQL database system also contains APIs for integration with c , c++ , PHP , java , python and ruby etc. You can connect it easily with different development platforms so you can actually integrate applications developed in different OS and development platforms.

- **Security**

MYSQL as a relational database is secure as all access passwords are stored in an encrypted format restricting any unauthorised access to the system. It also encrypts the transaction so eavesdroppers and data harvest tools cannot replicate or regenerate the database transactions once they are processed.

- **Connectivity**

MYSQL clients can access this relational database through standard TCP/IP sockets , named pipes , UNIX sockets and many more. Standard ODBC 2.5 and above functions and commands are also supported in MYSQL.

2.4 DATABASE SERVERS

A database server is used to store data in a database. Users can access the data and manipulate it. A web application can provide the user with the interface to the database. There are many types of databases. The most popular among them is the Relational Database Management System (RDBMS).

RDBMS:

RDBMS is a type of database management system that stores data in the form of related tables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

SQL:

The structured Query Language (SQL) comprises one of the fundamental building blocks of modern database architecture. SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, Oracle, DB2, Informix, MS SQL Server and Sybase etc.

A database most often contains one or more tables. Each table is identified by a name (E.g. “staff” or “traveller”). A table contains record (rows) with data. With SQL we can query a database and have a result set returned. SQL is the syntax for executing queries. But the SQL language also includes the syntax to insert and delete records. These query and update commands together form the Data Manipulation Language (DML) part of SQL. The Data Definition Language (DDL) part of SQL permits database tables to be created or detected. We can also define indexes (keys), specify links between tables and imposes constraints between databases.

2.5 FEASIBILITY STUDY

The key condition involved in the feasibility analysis are:-

□ Economical Feasibility.
□ Operational Feasibility.
□ Technical Feasibility.

Economical Feasibility:-

Economical feasibility analysis is the most frequently used evaluating technique for the effectiveness of the candidate system. These checks are their sufficient benefits in creating the system to make cost acceptable . In the case of the proposed system it is very necessary to implement in such a firm and when its necessity compared its cost for implementing is very low and it is very acceptable by the users of the system. When its advantages and efficiency are evaluated it is economically feasible.

Operational Feasibility:-

The proposed systems are beneficial only if they can turn into information systems that will meet the organization’s operating requirements. This test of feasibility asks if the system will work when it is developed and installed. Since this system is developed in vb.net its operation is very simple, very attractive and user friendly. The software is very much available, and it is the most used package now, and since it is a Microsoft version it provides many tools which helps to make usage very easier. It is very much acceptable by the users of the systems and it is keeping its standard.

Technical Feasibility:-

This checks that , can the work for the project be done with current equipment and existing software technology and available personnel. Since no other technical support is necessary for the function of the system we can conclude that it is technically feasible. When it is implemented there is only need of a computer (pc) and its other peripherals as usual so it is very satisfied in the matter of technical feasibility.

SYSTEM DESIGN

System design is the process of developing a candidate system that meets the criteria established in system analysis. User requirements are translated into system characteristics during system design. System design involves firstly the logical design and then physical construction of a new system. The logical design describes the detailed specifications for the new system, the input/output, files and databases, procedures, all in a manner that must be a project requirement. Physical construction, the activity following design produces software files and a working system.

3.1 INPUT DESIGN

Input design is a part of the overall system design, which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and people involved. If the data going into the system is incorrect, then the processing and output will magnify the errors. Thus the clear objectives of input design are:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable to and understood by the user.

3.2 OUTPUT DESIGN

The output design is done so that the result of processing could be committed to the user and to provide a hard copy of these results and evaluations for later consultations. Effective output design will improve the clarity and performance of outputs. Output design phase of the system is concerned with the convergence of information to the end user-friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and thereby enhancing the process of decision making. Outputs from the computer systems are required primarily to communicate the results of the processing to the users. They are also used to provide a permanent copy of these results of processing to the users.

They are also used to provide a permanent copy of these results for late consultation. There are various types of output required by most systems, the main ones are:

- External outputs, whose destination is outside the organization and which require special attention because they project the image of the organization.

- Internal outputs, whose destination is within the organization and which require careful design because they are the user’s main interface with the computer.
- Operational outputs, whose use is purely within the computer department.
- Turn around outputs, to which the data will be added before they are returned to the computer for further processing.

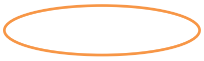
3.3 DATA FLOW DIAGRAM

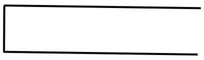
The data flow diagram (DFD) is one of the most important tools used by system analysts. Data flow diagrams are made up of a number of symbols which represent system components. Most data flow modelling methods used four kinds of symbols. These symbols are used to represent four kinds of system components. Processes, data source, data flows and external entities. Processes are represented by circles in DFD. Data flow represented by a thin line in the DFD and each data store has an unique name and square or rectangle represents external entities

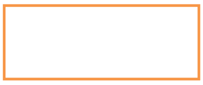
Unlike detailed flow charts, data flow diagrams do not supply detailed descriptions of the modules but graphically describes a systems data and how the data interact with the system

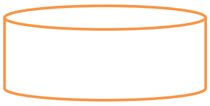
To construct a data flow diagram, we use

□ Arrow

□ Circle 

□ Open end box 

□ Square 

□ Database 

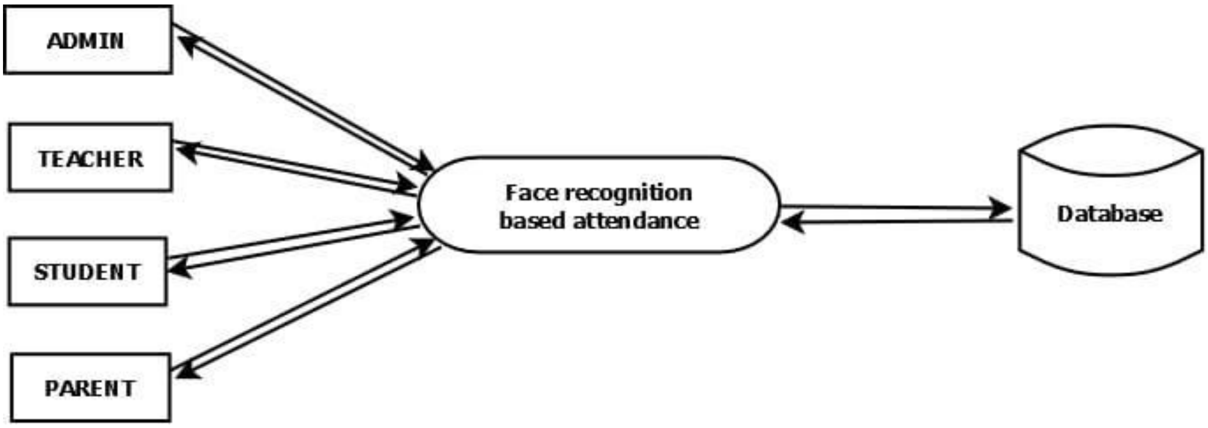
An arrow identifies the data flow in motion. It is a pipeline through which information is flown like the rectangle in the flow chart. A circle stands for a process that converts data into information. An open ended box represents a data store, data at rest or a temporary repository of data. A square defines a source or destinations of system data.

Five rules for constructing a data flow diagram

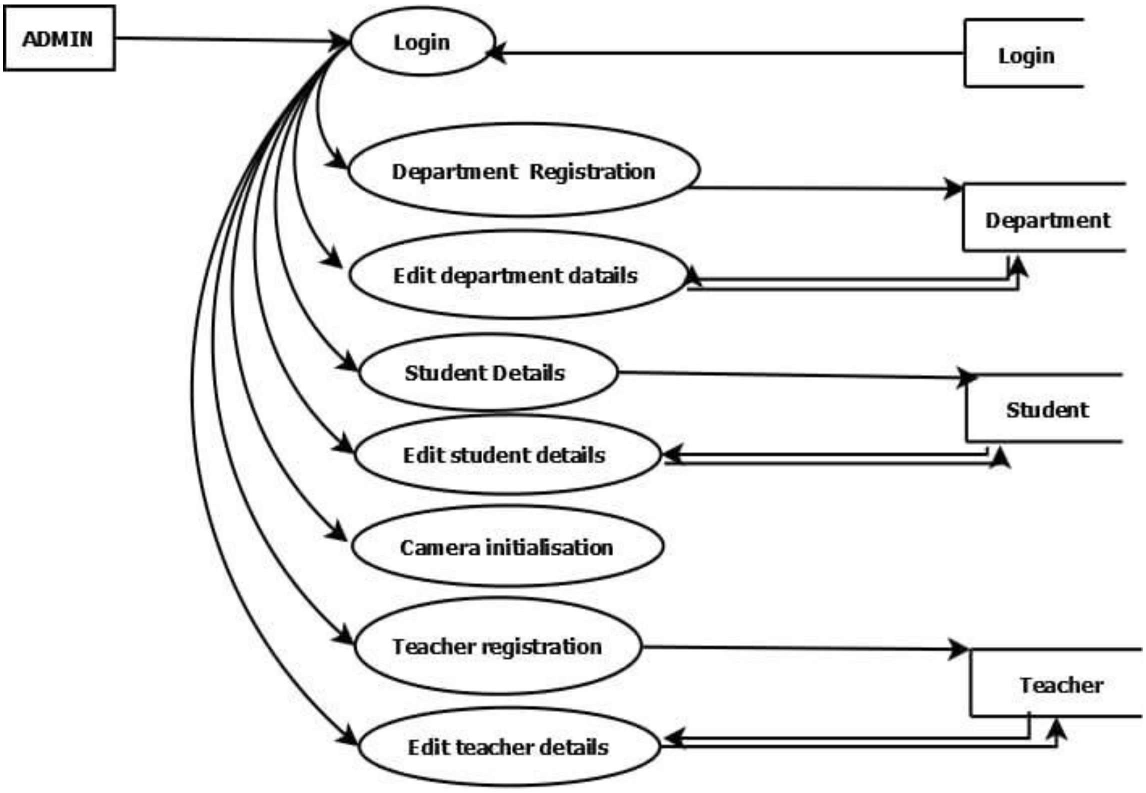
- Arrows should not cross each other
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have the same names.
- Choose meaning full data flow.
- Draw all data flows around the outside of the diagram

DATA FLOW DIAGRAM OF PROJECT

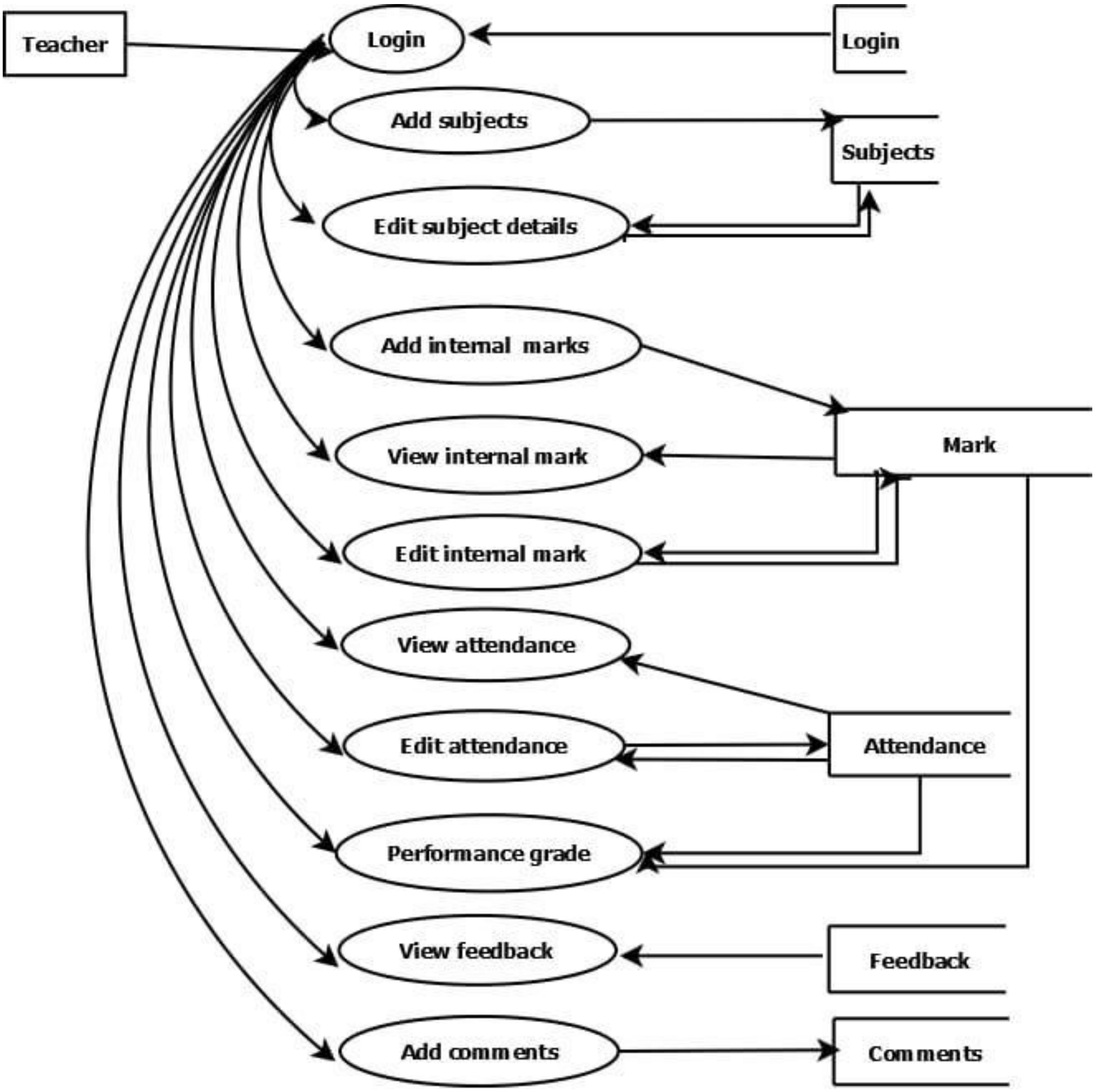
Level 0



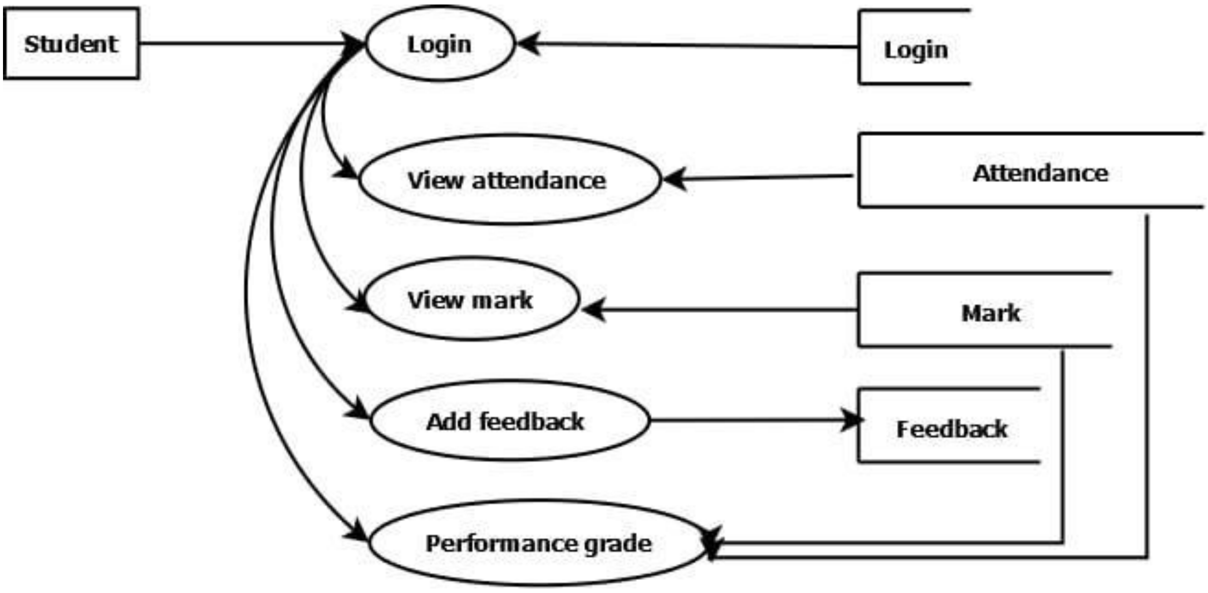
Level 1.1


ADMIN
Login
Login
Department Registration
Department
Edit department datails
Student Details
Student
Edit student details
Camera initialisation
Teacher registration
Teacher
Edit teacher details

Level 1.2

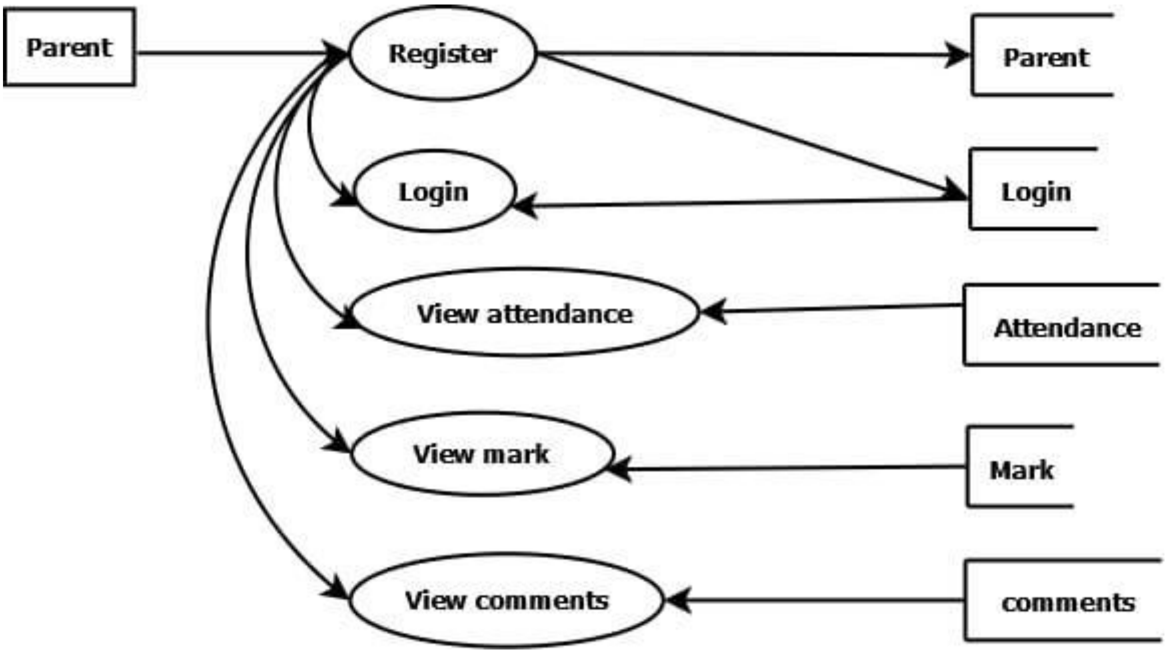

Teacher
Login
Login
Add subjects
Subjects
Edit subject details
Add internal marks
Mark
View internal mark
Edit internal mark
View attendance
Edit attendance
Attendance
Performance grade
View feedback
Feedback
Add comments
Comments

Level 1.3



Student
Login
Login
View attendance
Attendance
View mark
Mark
Add feedback
Feedback
Performance grade

Level 1.4


Parent
Register
Parent
Login
Login
View attendance
Attendance
View mark
Mark
View comments
comments

3.4 ENTITY RELATIONSHIP DIAGRAM

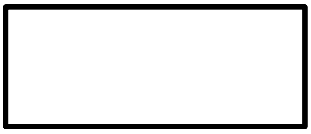
An entity relationship diagram is a data modelling technique that creates a graphical representation of the entities and the relationship between entities within an information system. An ER model is an abstract way to describe a database. Describing a database usually starts with a relational database, which stores data in tables. Some of the data in these tables point to data in the other tables.

An entity may be defined as a thing which is recognized as being capable of an independent existent and this can be uniquely identified. An entity is an abstraction from the complicity of a domain. When we speak of an entity, we normally speak of some aspect of the real world which can be distinguished from other aspects of the real world.

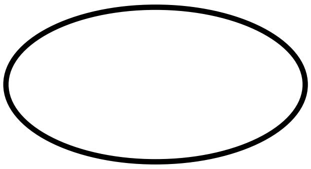
Three basic elements in ER model include:

- Entities are the things about which we seek information
- Attributes are the data we collect about the entities.
- Relationships provide the structure needed to draw information from multiple entities.

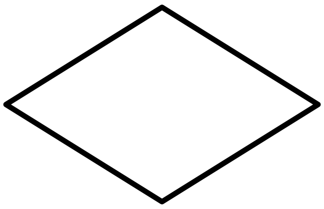
To construct an ER diagram, we use:



: Entity

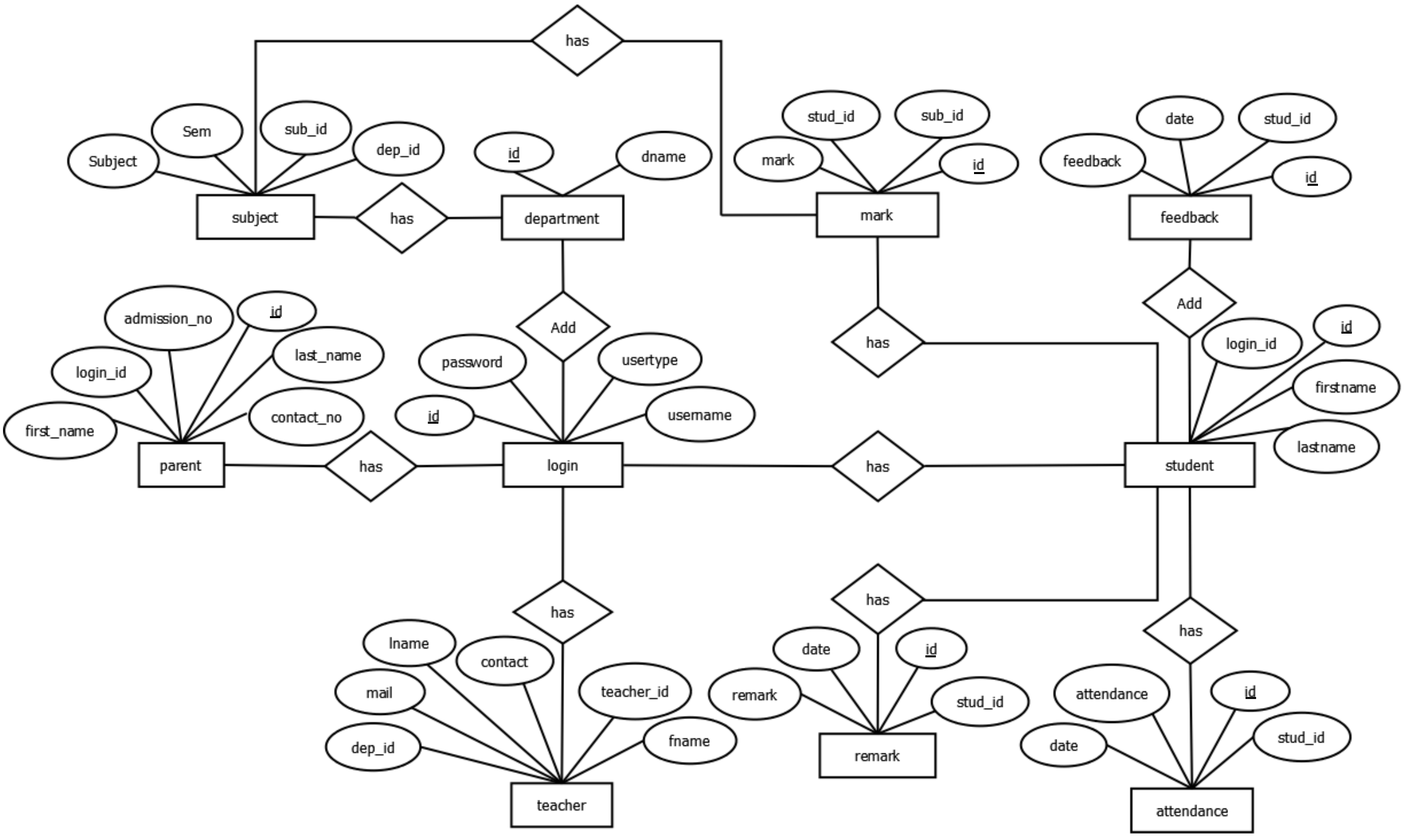


: Attribute



: Relation

: Connection


has
stud_id
sub_id
date
stud_id
Sem
sub_id
dep_id
Subject
id
dname
mark
id
feedback
id
subject
has
department
mark
feedback
admission_no
id
Add
has
Add
last_name
password
usertype
login_id
id
login_id
firstname
first_name
contact_no
id
username
lastname
parent
has
login
has
student
has
has
has
lname
contact
date
id
mail
teacher_id
remark
stud_id
attendance
id
fname
dep_id
date
stud_id
remark
teacher
attendance

3.5 TABLE DESIGN

NORMALIZATION

Normalization is the process of decomposing a set of relations with anomalies to produce smaller and well-structured relations that contain minimum redundancy. It is a formal process of deciding which attributes should be grouped together in a relation.

First Normal Form

First Normal form (1NF) is now considered to be part of the formal definition of relational model.1NF is designed to disallow multivalve attributes, composite attributes, and their combinations. It states that the domain of an attribute must include only atomic values. A domain is atomic, if elements of the domain are considered to be indivisible units. We say that a relational schema R is in 1NF if the domain of all attributes of R is atomic.

Second Normal Form

Second Normal form (2NF) is based on the concept of functional dependency. A relation R is in 2NF if it is in 1NF and every non key attribute A of R is fully dependent on the primary key. That is, relation is said to be in 2NF if each attribute An in R meets one of the following criteria:

(a) It appears in the primary key. (b) It is fully functionally dependent on the primary key.
The tables designed in the proposed system contain a primary key for uniquely identifying each user.

Third Normal Form

Third Normal form (3NF) is based on the concept of transitive dependency. A relation is said to be in 3NF if it is in 2NF and has no transitive dependencies. That is all the non key attributes should be functionally determined by the primary key. In the proposed system all attributes of tables fully depend on the primary key only, that is all non-key attributes are mutually independent.

DESIGN

TABLE_LOGIN

Field name	Data type	Constraints
Id	int	Primary key
Username	Varchar(50)	Not null
Password	Varchar(50)	Not null
Usertype	Varchar(50)	Not null

TABLE_TEACHER

Field name	Data type	Constraints
Teacher_id	int	Primary key
Fname	Varchar(20)	Not null
Lname	Varchar(20)	Not null
Gender	Varchar(20)	Not null
Address	Varchar(30)	Not null
District	Varchar(30)	Not null
Place	Varchar(30)	Not null
Post	Varchar(20)	Not null
Pin	bigint	Not null
Contact number	bigint	Not null
Mail	Varchar(50)	Not null
Dept_id	int	Foreign key

TABLE_STUDENT

Field name	Data type	Constraints
Id	int	Primary key
login_id	int	Foreign key
First_name	Varchar(20)	Not null
Last_name	Varchar(20)	Not null
Admission_no	int	Not null
Dept_id	int	Foreign key
Semester	int	Not null
Photo	Varchar(100)	Not null
Date_of_birth	date	Not null
Gender	Varchar(20)	Not null
Address	Varchar(50)	Not null
District	Varchar(20)	Not null
Place	Varchar(20)	Not null
Post	Varchar(20)	Not null
Pin	bigint	Not null
Contact_number	bigint	Not null
Mail	Varchar(50)	Not null

TABLE_PARENT

Field name	Data type	Constraints
Id	int	Primary key
Login_id	int	Foreign key
First_name	Varchar(20)	Not null
Last_name	Varchar(20)	Not null
Admission_no	int	Not null
Contact_no	bigint	Not null

TABLE_DEPARTMENT

Field name	Data type	Constraints
Id	int	Primary key
Department_name	Varchar(30)	Not null

TABLE_SUBJECT

Field name	Data type	Constraints
Sub_id	int	Primary key
Dep_id	int	Foreign key
Semester	int	Not null
Subject	Varchar(30)	Not null

TABLE_ATTENDANCE

Field name	Data type	Constraints
Id	int	Primary key
Stud_id	int	Foreign key
Date	datetime	Not null
Attendance	double	Not null

TABLE_MARK

Field name	Data type	Constraints
Id	int	Primary key
Stud_id	int	Foreign key
Sub_id	int	Foreign key
Mark	double	Not null

TABLE_FEEDBACK

Field name	Data type	Constraints
Id	int	Primary key
Stud_id	int	Foreign key
Date	date	Not null
Feedback	Varchar(50)	Not null

TABLE_REMARK

Field name	Data type	Constraints
Id	int	Primary key
Stud_id	int	Foreign key
Date	Varchar(50)	Not null
Remark	Varchar(50)	Not null

SYSTEM TESTING

4.1 TESTING

Testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Thus the system test should be a confirmation that all is correct and an opportunity to show the users that the system works. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is the process of executing a program with the explicit; it can only show that software defects at present.

System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

The testing steps are:

- Unit testing
- Integration testing
- Validation
- Output testing
- User acceptance testing

UNIT TESTING

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “module testing”. The modules are tested separately. This testing is carried out during the programming stage itself. In this testing step each module is found to be working satisfactorily as regard to the expected output from the module

INTEGRATION TESTING

Integration testing focuses on the design and the construction of the software architecture. Data can be lost across an interface; one module can have adverse effects on another sub function and so on. Thus integration testing is a part that the software meets all functional, behavioural and performance requirements. The errors, which are uncovered during integration testing, are corrected during this phase.

Integration testing systematic techniques for conducting the program structure. While at the same time conducting tests to uncover errors associated with the interfacing.

VALIDATION TESTING

Errors discovered where corrected prior to completion of this project with the help of the user by negotiating to establish a method of resolving deficiencies. Thus the proposed system under validation testing and found to be working satisfactorily. In validation testing the requirements established as a part of software requirements are validated against the software that has been constructed. Validation testing provides the final assurance that software meets all functional behavioural and performance requirements.

OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. The outputs generated or displayed by the system under consideration are tested asking the users about the format required by them. Here, the output is considered in two ways: one is the screen and the other is printed format.

The output format on the screen is found to be correct as the format designed according to the users need. For the hard copy also, the output comes out as specified by the user. Hence output testing does not result in any correction in the system.

USER ACCEPTANCE TESTING

Acceptance testing includes final testing of the complete system to user satisfaction and supervision of the new system.

User acceptance testing is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. This is done with regard to the following points.

- Input screen design
- Process screen design
- Selecting avi and txt file
- Return to the code directory

The above testing is done by taking various kinds of data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system by using test data errors are given uncovered and corrected by using above testing and corrections are also noted for future use

4.2 PROCESS TESTING

The test team consists of the company staff, the user, and myself. The company staff are very experienced in the testing field. We test the system by running it in several ways by inputting various The main concerns in our testing are following:

- Browser compatibility: This test validates consistent applications performance on a variety of browser types and configurations.
- Functional correctness: These tests validate that the application functions correctly.
- Integration: These test the integration between browsers and servers, applications and data, hardware and software.
- Usability: These test the overall usability of a web page or a web application, including appearance clarity and navigation.
- Security: These test the adequacy and correctness of security controls including access control and authorizations.
- Performance: These test the performance of the web applications under load.
- Verification of code: This validates that the code used in building the web application has been used in a correct manner.

SYSTEM IMPLEMENTATION

System Implementation

The **implementation** phase of a project covers the period from the acceptance of the tested design to its satisfactory operations, supported by the appropriate user and the operation manual. It is a major operation across the whole organization structure and requires a great deal of planning. Planning of implementation must begin from initial conception of the project. It requires a thorough knowledge of the new system, its personal need, hardware and software requirements, file and procedure conversion activities and of the current system where interface with the new, the change to it, the job that will be superseded, etc. Only the analyst responsible for creation of the new system will possess this knowledge. The new system analyst can plan, schedule and coordinate, but has no executive power. Planning must cover the following aspects.

- Organization of implementation.
- Control of resources.
- Motivation of the users.
- Training and production manuals.
- Change over.

CONCLUSION

Conclusion

This package has been developed to handle the student attendance system.. This system is designed using a generalized application and is also a highly user-friendly one.

The system is more effective. Less time and paperwork is required. No chance of error. Users can generate the report as per requirement or in the middle of the session. Students can improve their attendance. Work can be done speedily and in time.

FUTURE ENHANCEMENT

Future Enhancement

We have so many limitations to implement all the ideas that we have found because of time limitations. So in future our system can be expanded with following properties

- Wireless webcam can be used.
- Make it suitable for all colleges.
- Fast face matching.

8.BIBLIOGRAPHY

Websites:-

1. http://www.w3schools.com

2. http://www.stackoverflow.com

3. http://www.sololearn.com

4. http://www.tutorialspoint.com

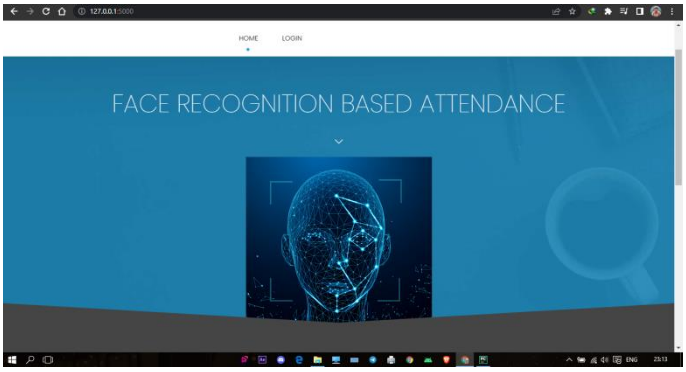
5. http://www.geeksforgeeks.org

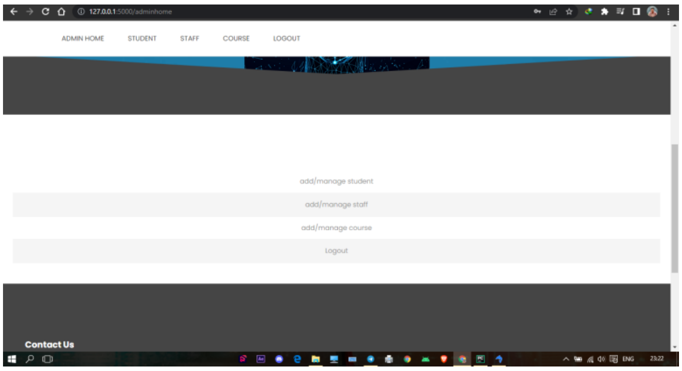
Reference Made From:

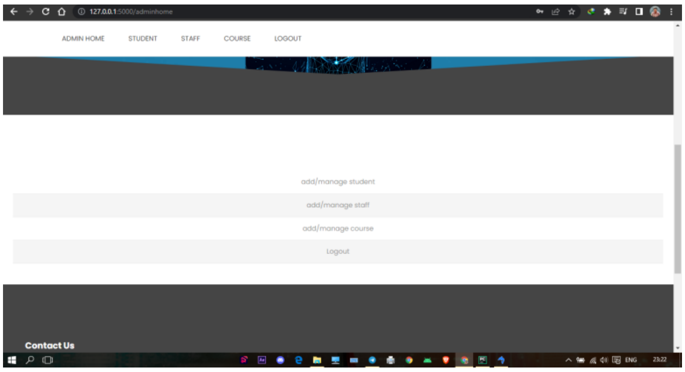
1. Test driven development with python by Harry Percival.

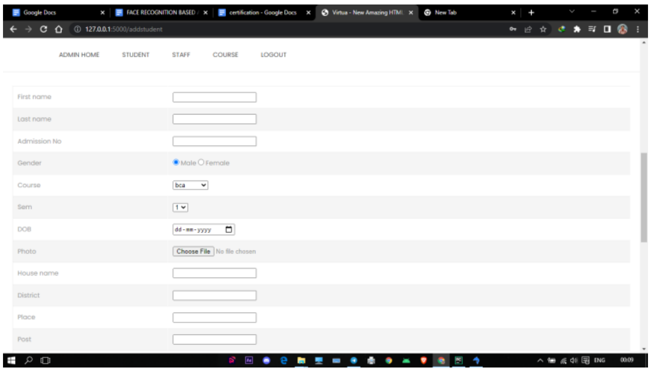
SCREENSHOTS

WEBSITE

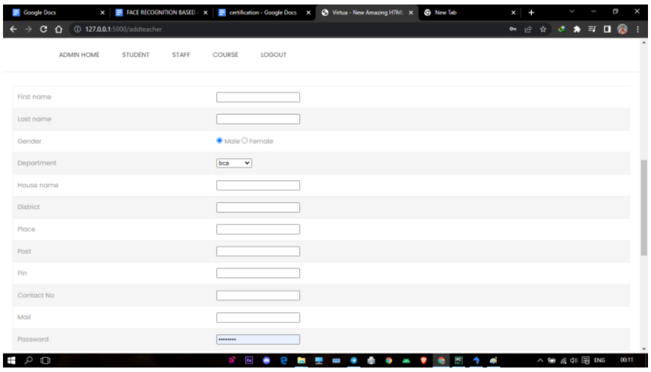




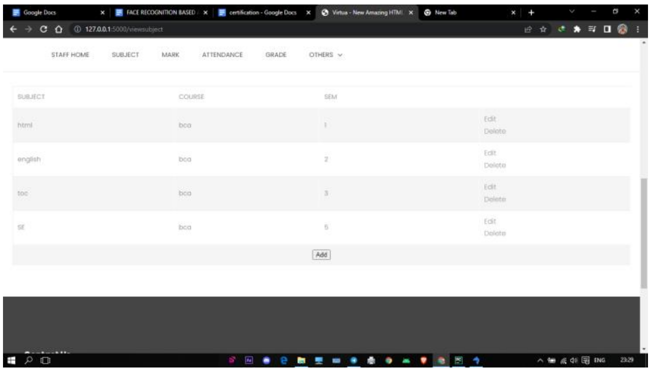




First name	
Last name	
Admission No	
Gender	● Male ○ Female
Course	bca
Sem	1
DOB	dd-mm-yyyy
Photo	Choose File No file chosen
House name	
District	
Place	
Post	

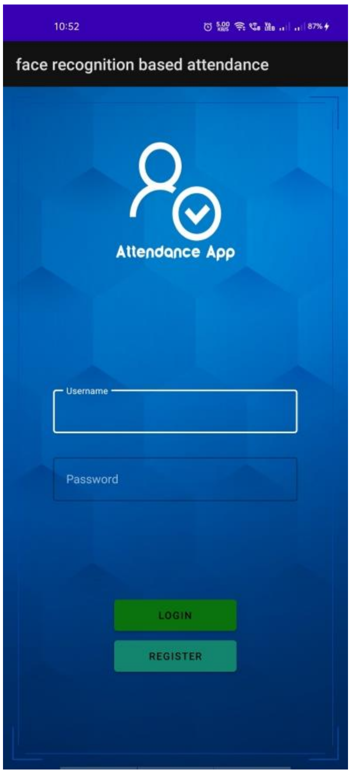


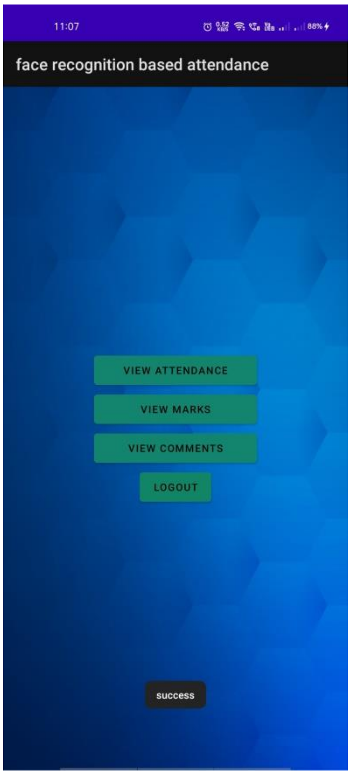
First name	
Last name	
Gender	● Male ○ Female
Department	bca
House name	
District	
Place	
Post	
Pin	
Contact No	
Mail	
Password	••••••••



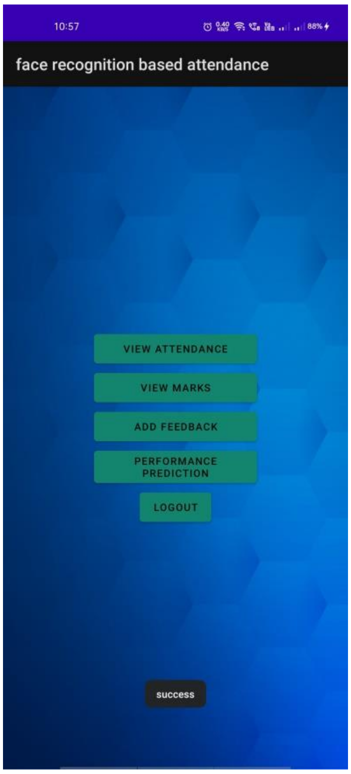
SUBJECT	COURSE	SEM	
html	bca	1	Edit Delete
english	bca	2	Edit Delete
toc	bca	3	Edit Delete
SE	bca	5	Edit Delete

ANDROID

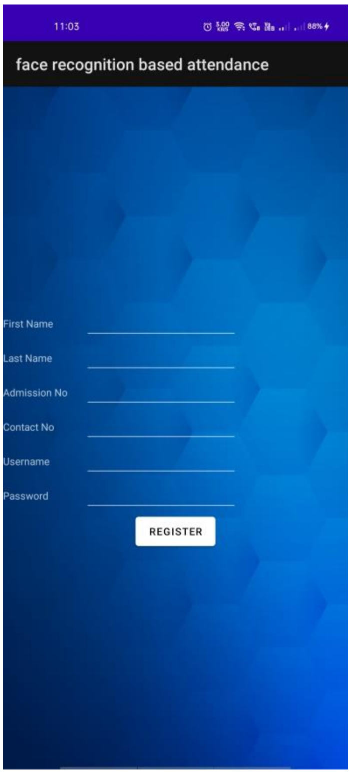


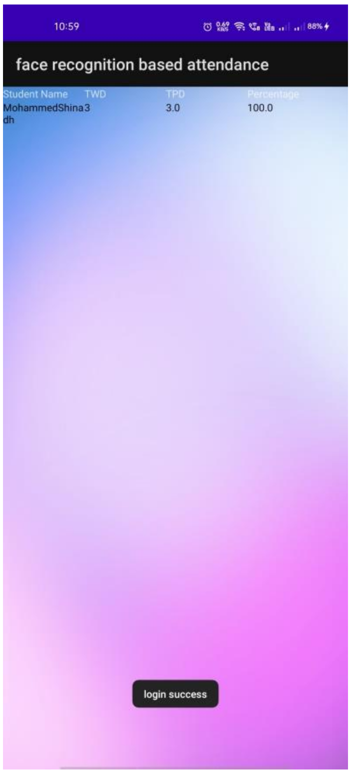


success



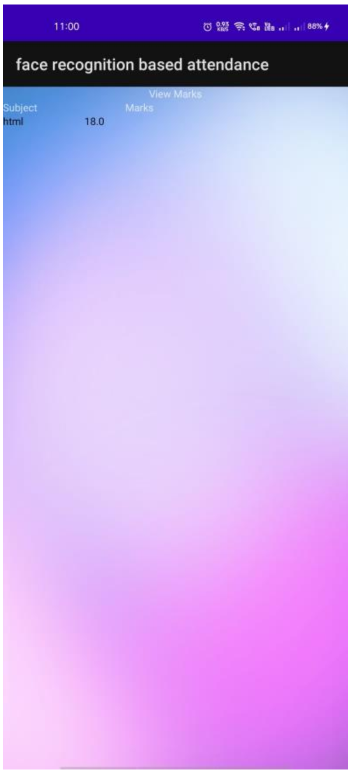
success





Student Name	TWD	TPD	Percentage
MohammedShinadh	3	3.0	100.0

login success



View Marks

Subject	Marks
html	18.0

SAMPLE CODES

WEBCODE.PY

```
from flask import *
from werkzeug.utils import secure_filename

from src.dbconnection import *

app=Flask(__name__)
app.secret_key="abc"
@app.route('/')
def main():
    # return render_template("index.html")
    return render_template("login.html")

@app.route("/loginnew",methods=['post'])
def loginnew():
    uname=request.form['textfield']
    passwd=request.form['textfield2']
    qry="select * from login where username=%s and password=%s"
    val=(uname,passwd)
    res=selectonenew(qry,val)
    print(res,"===========")
    if res is None:
        return'''<script>alert("invalid");window.location="/"</script>'''
    elif res[3]=="admin":
        return '''<script>alert("login
success");window.location="/adminhome"</script>'''
    elif res[3]=="teacher":
        return'''<script>alert("login
success");window.location="/staffhome"</script> '''
    else:
        return'''<script>alert("invalid");window.location="/"</script>'''
```

LOGIN.HTML

```
{% extends 'index.html' %}
{% block body %}
<form action="loginnew" method="post">
 <div align="center">
   <table width="262" border="0">
     <tr>
       <td width="170">USERNAME</td>
     <td width="76"><input type="text" name="textfield" required/></td>
   </tr>
     <tr>
       <td>PASSWORD</td>
     <td><input type="password" name="textfield2" required/></td>
   </tr>
     <tr>
       <td>&nbsp;</td>
     <td><input type="submit" name="Submit" value="LOGIN" /></td>
   </tr>
   </table>
 </div>
</form>
{% endblock %}
```

STUDENTHOME.JAVA

```
package com.example.facerecognitionbasedattendance;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class studenthome extends AppCompatActivity {
Button b8,b9,b10,b11,b14;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_studenthome);
       b8=findViewById(R.id.button8);
       b9=findViewById(R.id.button9);
       b10=findViewById(R.id.button10);
       b11=findViewById(R.id.button11);
       b14=findViewById(R.id.button14);
       b8.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               Intent i=new
Intent(getApplicationContext(),ViewAttendance.class);
               startActivity(i);
           }
       });
       b9.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               Intent i=new
Intent(getApplicationContext(),viewmarks.class);
               startActivity(i);
           }
       });
       b10.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               Intent i=new
Intent(getApplicationContext(),addfeedback.class);
               startActivity(i);
           }
       });
       b11.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               Intent i=new
Intent(getApplicationContext(),performanceprediction.class);

               startActivity(i);

           }

       });

       b14.setOnClickListener(new View.OnClickListener() {

           @Override

           public void onClick(View view) {

               Intent i=new
Intent(getApplicationContext(),MainActivity.class);

               startActivity(i);

           }

       });

   }
```