

# **FACIAL AND HAND GESTURE BASED MEDIA PLAYER**

*A project report submitted to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY-H, HYDERABAD**

*In partial fulfillment of the requirements  
for the award of the degree of*

**BACHELOR OF TECHNOLOGY  
IN  
COMPUTER SCIENCE AND ENGINEERING**

*Submitted by  
IV B.Tech II Semester*

<b>P. THIRUPATHI</b>	<b>(16UC1A0539)</b>
<b>B. VANI</b>	<b>(16UC1A0514)</b>
<b>D. VAISHNAVI</b>	<b>(16UC1A0520)</b>
<b>S. SONY</b>	<b>(16UC1A0547)</b>

*Under the Esteemed Guidance of*  
**Prof. Dr. A. Kiran Mayee, B.E., PH.D.**  
**Prof & Principal**



**Department of Computer Science and Engineering**  
**TALLA PADMAVATHI COLLEGE OF ENGINEERING**

**(Affiliated to JNTUH, Hyderabad and approved by AICTE, New Delhi)**  
**Somidi, Kazipet, Warangal Urban-506003.**

**(2016-2020)**



## TALLA PADMAVATHI COLLEGE OF ENGINEERING

(Affiliated to JNTUH, Hyderabad and approved by AICTE, New Delhi)  
Somidi, Kazipet, Warangal Urban, Telangana-506003.

Department of Computer Science and Engineering

# Certificate

This is to certify that the project work entitled

**“FACIAL AND HAND GESTURE BASED MEDIA PLAYER”**

is the bonafide work done by

<b>P. THIRUPATHI</b>	<b>(16UC1A0539)</b>
<b>B. VANI</b>	<b>(16UC1A0514)</b>
<b>D. VAISHNAVI</b>	<b>(16UC1A0520)</b>
<b>S. SONY</b>	<b>(16UC1A0547)</b>

In the Department of Computer Science and Engineering, **Talla Padmavathi College of Engineering**, Somidi is affiliated to JNTU-Hyderabad in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2016-2020.

This work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

### INTERNAL GUIDE:

**DR. A. KIRAN MAYEE B.E, PH.D.**  
PROFESSOR & PRINCIPAL  
DEPT OF CSE  
T.P.C.E  
SOMIDI, WARANGAL URBAN

### HEAD OF DEPARTMENT:

**ASST.PROF. K. DEEPIKA**  
ASSISTANT PROFESSOR & HEAD  
DEPT OF CSE  
T.P.C.E  
SOMIDI, WARANGAL URBAN

INTERNAL EXAMINER

EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

Before getting into the thickest of things, we would like to thank the personalities who were part of my project in numerous ways, those who gave me outstanding support from birth of the project.

We are extremely thankful to our beloved chairman and establisher **Talla Mallesham** & Managing Directors **Mr. Talla Vamshi** and **Mrs. Talla Chaithanya** for providing necessary infrastructure and resources for the accomplishment of our project at **Talla Padmavathi College of Engineering, Warangal Urban.**

We are highly indebted to **Prof. Dr. A. Kiran Mayee, Principal of Talla Padmavathi College of Engineering**, for his support during the tenure of the project.

We are very much obliged to our beloved **Asst. Prof. K. Deepika, Head of the Department of Computer Science & Engineering**, Talla Padmavathi College of Engineering for providing the opportunity to undertake this project and encouragement in completion of this project.

We hereby wish to express our deep sense of gratitude to **Prof. Dr. A. Kiran Mayee, Principal**, Department of Computer Science and Engineering, Talla Padmavathi Engineering College for the esteemed guidance, moral support and invaluable advice provided by him for the success of the project.

We are also thankful to all the staff members of Computer Science and Engineering department who have co operated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Thanks for Your Valuable Guidance and kind support.

# DECLARATION

We hereby declare that project report entitled **“FACIAL AND HAND GESTURE BASED MEDIA PLAYER”** is a genuine project work carried out by us, in **B.Tech (Computer Science and Engineering)** degree course of **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**

and has not been submitted to any other courses or University for award of any degree by us.

## Signature of the Student

- 1.
- 2.
- 3.
- 4.

# **ABSTRACT**

Advance technologies are the key factor of today's upcoming world, in this project we are developing an advanced media player which plays and pauses the video by detecting the users face looking at screen or not. System monitors whether the user is looking at the screen or not using a web camera. If yes, then doesn't interrupts the video and allows it to play. In case if the user is not looking at the or say the system couldn't detect the users face then it immediately stops the video.

We are trying to add a feature of controlling other features of media player such as volume up and volume down, backward and forward video using hand gestures. Hand gestures plays a crucial role in increasing and reducing the volume. Look based media player also work as the field of computer vision based on hand gesture interfaces for Human Computer Interaction (HCI).

Currently we have proposed to build a prototype which gives an accurate result in terms of performance. This media player also helps the users to have better experience in daily life. It also helps with the user friendly performance.

# CONTENTS

Chapter	Name of the Chapter	Page
<b>1.</b>	<b>Introduction</b>	<b>01</b>
	1.1 Introduction	02
	1.2 Statement of the problem	03
	1.3 Objectives	03
	1.4 Scope	03
	1.5 Purpose	04
	1.6 Limitations	04
<b>2.</b>	<b>Literature Survey</b>	<b>05</b>
	2.1 Information retrieval	06
	2.2 Detection Methods	07
	2.3 Recognition Methods	09
<b>3.</b>	<b>Analysis</b>	<b>11</b>
	3.1 Existing System	12
	3.2 Disadvantages	12
	3.3 Proposed system	12
	3.4 Advantages	13
	3.5 System Used	13
<b>4.</b>	<b>Design</b>	<b>14</b>
	4.1 UML Diagrams	15
<b>5.</b>	<b>Implementation</b>	<b>19</b>
	5.1 Face Detection	20

	5.2 Hand Gesture	21
	5.3 Media Player	22
<b>6.</b>	<b>Testing</b>	<b>23</b>
	6.1 Types of tests	24
	6.2 Test strategy and approach	26
	6.3 Test results	26
<b>7.</b>	<b>Results and Performance Evaluation</b>	<b>27</b>
	7.1 Results	28
	7.2 Performance Evaluation	28
<b>8.</b>	<b>Conclusion and Future work</b>	<b>29</b>
<b>9.</b>	<b>Appendix</b>	<b>31</b>
	9.1 Object Oriented Analysis and Design through Unified Modeling Language	32
	9.2 Software Environment	36
	9.3 Coding	41
	9.4 List of Figures	48
	9.5 List of Tables	49
	9.6 List of Abbreviations	50
	9.7 Screen Shots	51
<b>10.</b>	<b>References</b>	<b>71</b>

## **1. INTRODUCTION**



## 1. INTRODUCTION

While watching a video when someone interrupts you and you have to look somewhere else or go away from the system for some time so you need to drag back the video from where you left. Well we got a solution to this problem. A media player that gets paused when user is not looking at it. The media player resumes again as soon as the user looks at it again. This can be done using the web camera. The media player will be played continuously as long as the camera detects the users face.

The media player pauses as soon as users face is not completely detected. This system also provides the feature of controlling other functions of media players such as play, pause, volume up, volume down, next using hand gestures helps to change to a new video.

Direct use of hands as an input device is an attractive method for improving natural Human Computer Interaction which has evolved from text based interfaces from 2D graphical based interfaces, multimedia supported interfaces, to fully fledged multi-participant Virtual Environment systems. This enhanced media player can help in minimizing human efforts. In future, this technique can Get better experience of using media player by Not missing any part of video etc.

This task of hand gesture recognition is one the important and elemental problem in computer vision. With recent advances in information technology and media, automated human interactions systems are build which involve hand processing task like hand detection, hand recognition and hand tracking.

This prompted my interest so I planned to make a software system that could recognize human gestures through computer vision, which is a sub field of artificial intelligence. The purpose of my software through computer vision was to program a computer to “understand” a scene or features in an image.

### **1.2 STATEMENT OF THE PROBLEM**

“Face and Hand Gesture Recognition Using Camera” is based on concept of Image processing. In recent year there is lot of research on gesture recognition using Kinect sensor on using HD camera but camera and Kinect sensors are more costly. This paper is focus on reduce cost and improve robustness of the proposed system using simple web camera.

Gesture recognition has been adapted for various research applications from facial gestures to complete bodily human action. Several applications have emerged and created a stronger need for this type of recognition system. Static gesture recognition is a pattern recognition problem; as such, an essential part of the pattern recognition pre-processing stage, namely, feature extraction, should be conducted before any standard pattern recognition techniques can be applied. Features correspond to the most discriminative information about the image under certain lighting conditions.

### **1.3 OBJECTIVES**

First objective of this project is to create a complete system to detect, recognize and interpret the hand gestures and face through computer vision to pause the video.

Second objective of the project is therefore to provide a new low-cost, high speed and color image acquisition system.

## **1.4 SCOPE**

This enhanced media player can help in minimizing human efforts. In future, this technique can be used to control system using HCI like pdf reader, power point etc. we are doing this by using face detection and hand gestures for controlling varied features of the media player.

## **1.5 PURPOSE**

The aim of the project is to monitor and control the media player. This media player can increase the level of human computer interaction (HCI) than the current scenario.

## **1.6 LIMITATIONS**

- System may not work in less light up work space.
- For face detection complete face must be visible

## **2. LITERATURE SURVEY**

## **2 LITERATURE SURVEY**

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, the next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need a lot of external support. This support can be obtained from senior programmers, from books or from websites. Before building the system the above considerations are taken into account for developing the proposed system.

### **2.1 Information Retrieval:**

Many researchers have proposed numerous methods for hand gesture recognition systems. Generally, such systems are divided into two basic approaches namely glove – based and vision – based approaches. In glove based analysis, detection of the hand is eliminated by the sensors on the hand and 3D model of the hand is easily subjected to the virtual world and analysis comes next. Such systems are optimal for body motion capture purposes and widely used in industry. On the other hand vision-based analysis is more natural and useful for real time applications.

A healthy human can easily identify a hand gesture, however for a computer to recognize hand gesture first the hand should be detected in the acquired image and recognition of that hand should be done in a similar way as humans do. Yet this is a more challenging approach to implement because of the limitations of such a natural system. The vision-based approaches are carried out by using one or more cameras to capture and analyze 2D or 3D shapes of hands.

Detection and gesture analysis of the hands is a growing literature topic and has many user environment limitations for most of the studies. Segmentation of the hand is the first step of such systems. Exceptionally, such systems in [10], [12] and [13] gloves or finger marks have been used to extract the hand posture information in the frame and ease the hand segmentation process by eliminating the varying skin color issue of the problem. This technique allows the system to detect hands in a straightforward manner and it is more robust to change in lightning conditions and it is also independent of the user's skin color.

## **2.2 Detection Methods:**

In recent years, with the introduction of a new approach [33], which has a high detection rate, new studies are mostly concentrated on Boosting and HMM. The most tempting side of using those methods is that they usually work with gray scale images instead of colored images and thus it eliminates the drawbacks of such color based noise issues. This innovative approach is using a well-known technique namely Adaboost classifier which was mentioned in [32]. Adaboost classifier is an effective tool to select appropriate features for face detection. This feature extraction technique does not need skin color information and have less computation time with the use of integral image concept. But the drawback of this method is that it requires a training process. This process often needs huge-sized sample images to have a high detection rate.

By considering the drawbacks of training based methods and since there is a huge amount of studies based on training in literature, a feature invariant method was chosen to locate faces in this study. Also the main of this thesis study is recognizing hands and detection of face is just

Approach	Instance Study
<u>Feature Invariant Methods</u> <ul style="list-style-type: none"> <li>● Facial Features</li> <li>● Texture</li> <li>● Skin Color</li> <li>● Multiple Features</li> </ul>	Facial Components Analysis [21] Gray Scale Classification [23] Adaptive Gaussian Mixture Model [16] Skin color, shape analysis and facial components
<u>Template Matching Methods</u> <ul style="list-style-type: none"> <li>● Predefined Templates</li> <li>● Deformable Templates</li> </ul>	Shape Template Matching [35] Skin Active Shape Model for Face Alignment. [20]
<u>Appearance-Based Methods</u> <ul style="list-style-type: none"> <li>● Principal Components Analysis</li> <li>● Neural Network</li> <li>● Support Vector Machine (SVM)</li> <li>● Bayes Classifier</li> <li>● Hidden Markov Model (HMM)</li> <li>● Boosting and Ensemble</li> </ul>	Event Detection by Eigenvector Decomposition Eigenface Decomposition for Face Recognition[34] Motion Pattern Classification by Neural Networks SVM with Fisher Kernels [12] Dynamic BN for Gesture Recognition [18] Input-Output HMM for Gesture Recognition [19] Detection using Boosted Features.[22]

an intermediate tool. Investigation of a new technique for face detection would be tempting and Multiple Features in table 2.1 was a good choice for detection of face. Skin color is the central invariant feature of this study. It is indicated that human skin color is independent of human race and the wavelength of the exposed light [36]. This fact is also valid for the transformed color spaces of common video formats and thus skin color can be defined as a global skin color cloud in the color space and this cloud is called skin locus of that color space[3]. The thresholds of the skin locus is too large to extract current skin color in an input image correctly. Since shadows, illumination and pigmentation of human skin conditions would vary in a wide range, it is reasonable to adapt this general thresholds and narrow the skin locus for the current conditions. Skin locus is supposed to include all the skin pixels in an image with some other skin like pixels. Those false positive pixels should be eliminated by a fine skin segmentation.

According to face detection method introduced in [2], colored images are investigated in 2 steps namely coarse skin color segmentation and fine skin color segmentation. For coarse skin color segmentation fixed skin color thresholds in nRGB color space are used (skin locus).

## 2.3 Recognition Methods:

Many studies in literature use different skin locus thresholds for different color spaces to locate faces or hands in images. [25] starts with an RGB image and by apply a dimension reduction algorithm to propose its own skin locus in two dimension to compare its performance with HSV skin locus. [24] and [29] compare HSV/HSI,RGB,TSL and YcbCr color space skin locus performances. [26] starts the segmentation of skin color in YUV color space to have a quick result and then tune the current skin color with a quasi-automatic method which needs some user input. In [27], chrominance along with luminance information in YCbCr color space was used to segment the skin color for current conditions and histogram clustering is used for fine skin segmentation.

In feature invariant face detection methods skin segmentation is followed by face verification. Once the skin color is segmented in the obtained image, skin pixels are grouped by a region growing algorithm to extract blobs. Those blobs are investigated if they are face or not. Many methods have been proposed for this purpose in literature. The most simple one is that calculating height to width ratios of the segmented blobs [31]. Typically a frontal face's height to width ratio would be in a certain interval and this information would yield elimination of some blobs which are not definitely corresponding to real face in the image. Height to width ratio alone is not sufficient to detect the face blob because skin like pixels might still construct similar height to width ratio blobs. in [28] symmetry analysis and facial components analysis are introduced to detect faces. Detecting symmetric eyes, lips and nose would be the best method to recognize detect a face however it contributes an important computation time to the algorithm. [2] propose a method between the simplicity and the computational complexity of these two methods and presents blob's mismatch area method.



A clear summary of such algorithms are shown in table

Reference	Primary Method of Recognition	Number of Gestures Recognized	Background to Gesture Images	Additional Markers Required	Number of Training Images
39	Hidden Markov Models	97	General	Multi-colored gloves	400
5	Entropy Analysis	6	No	No	400
41	Linear approximation to non-linear point distribution models	26	Blue Screen	No	7441
42	Finite State machine modeling	7	Static	Markers on glove	10 sequences of 200 frames each
43	Fast Template Matching	46	Static	Wrist band	100 examples per gesture

In literature, if a gesture recognition system is constructed on a training based methods, then the number of gestures that will be recognized could be increased. However if it is an invariant method or state based method the number of gestures could not be increased easily. However, the system would be faster and independent of the training process.

### **3. ANALYSIS**

### 3. ANALYSIS

#### 3.1 Existing System

Mostly existing systems use eye recognition. Due to which results aren't accurate. Face recognition and hand gestures are not implemented properly together and not even individually.

#### 3.2 Disadvantages

There is a problem with the existing system. First, it involves a high computational cost, since we would have to repeat a large number of gestures similarity computations for every gesture.

#### 3.3 Proposed System:

In this project we are using face recognition and hand gestures for controlling media player. Face recognition is used for pausing and playing. Various hand gestures are used for controlling other functions of media player.

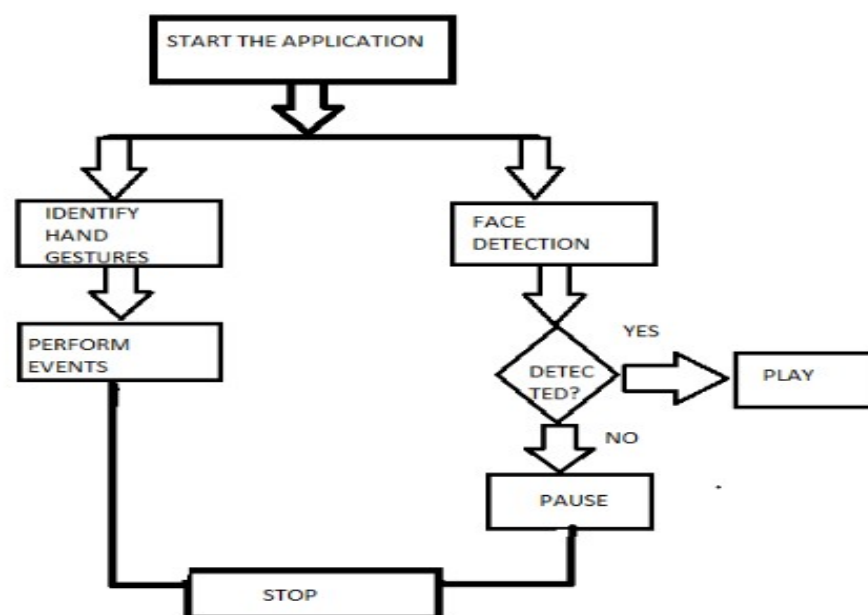


FIG: PROPOSED SYSTEM

### 3.4 Advantages

Facial and hand gesture based media player following advantages:

- Users cannot miss any part of the video.
- The video stops as user changes their view from the video thereby no need of users to keep on dragging back to the point from where they missed.
- You can also forward and backward the video if required.
- It saves time and electricity.
- It gives accurate result

### 3.5 System Used:

The following are the requirements we used to run proposed system.

#### Hardware System Configuration:

ANY CONTEMPORARY PC

#### Software System Configuration:

Operating System	□	UBUNTU / UNIX
Scripts	□	Python
Dependencies	□	lmutils, numpy, opencv-python, vlc-ctrl

## **4. DESIGN**

## UML DIAGRAMS:

### Class Diagram:

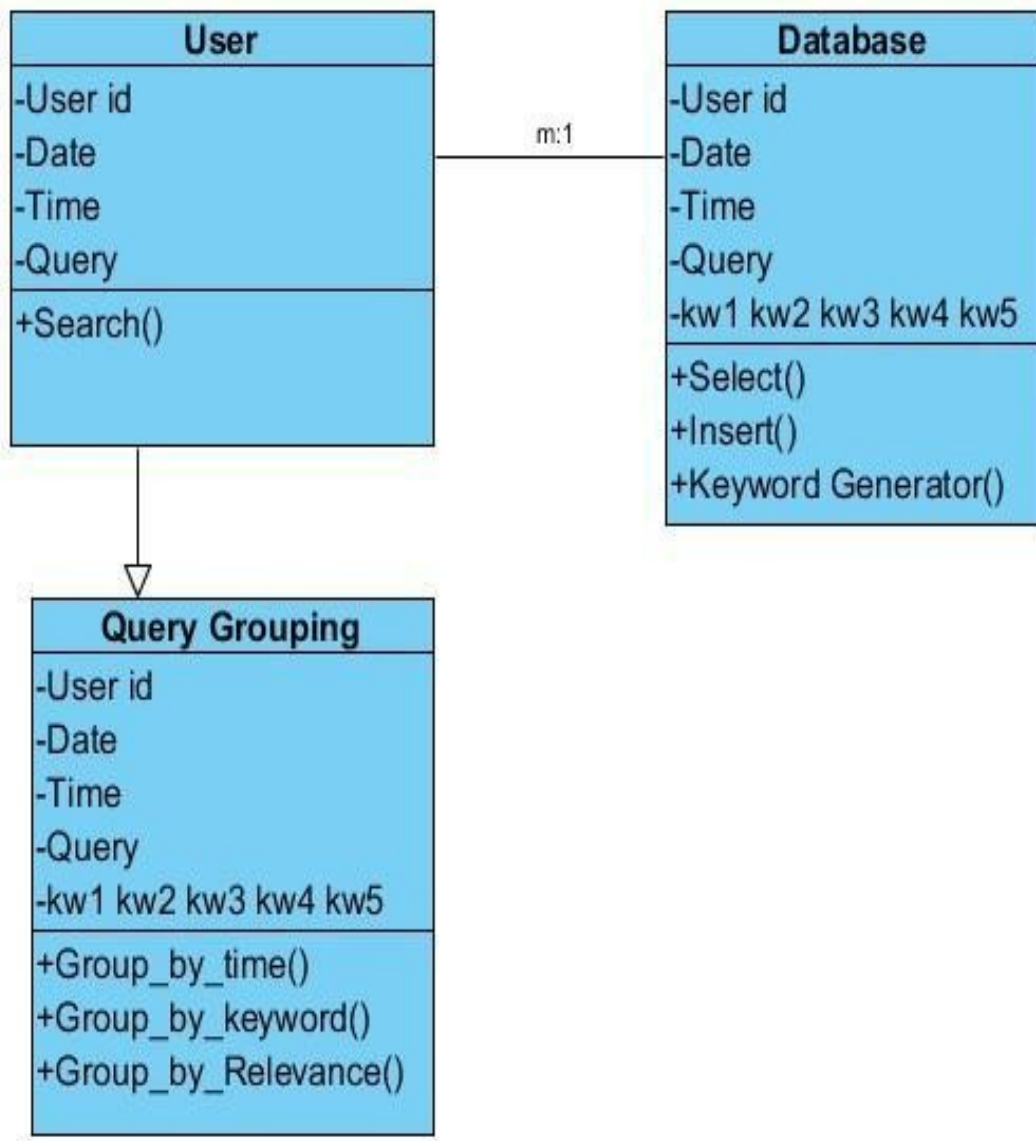


Fig.4.1. Class Diagram

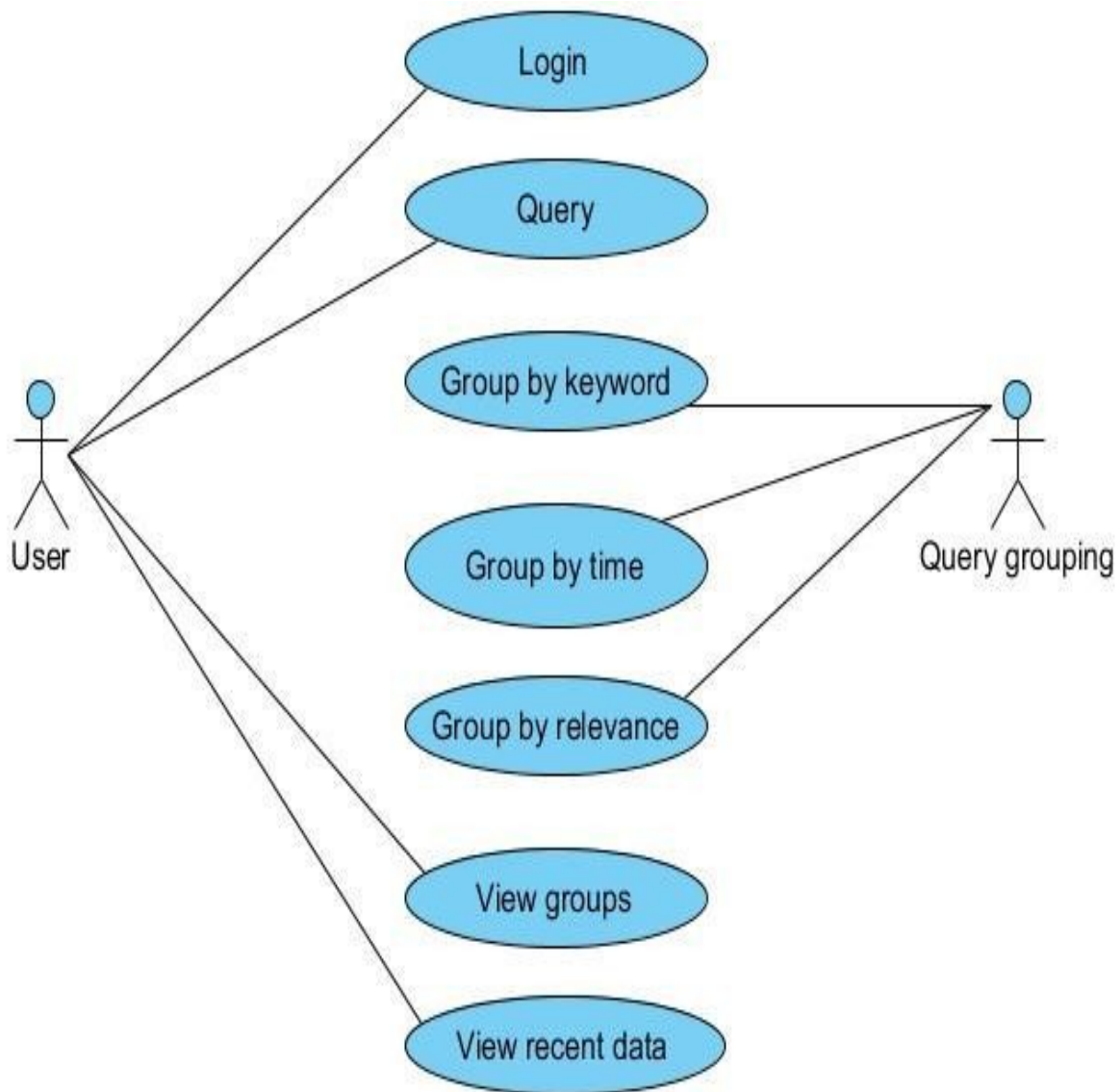
**Use Case Diagram:**

Fig.4.2. Use case Diagram

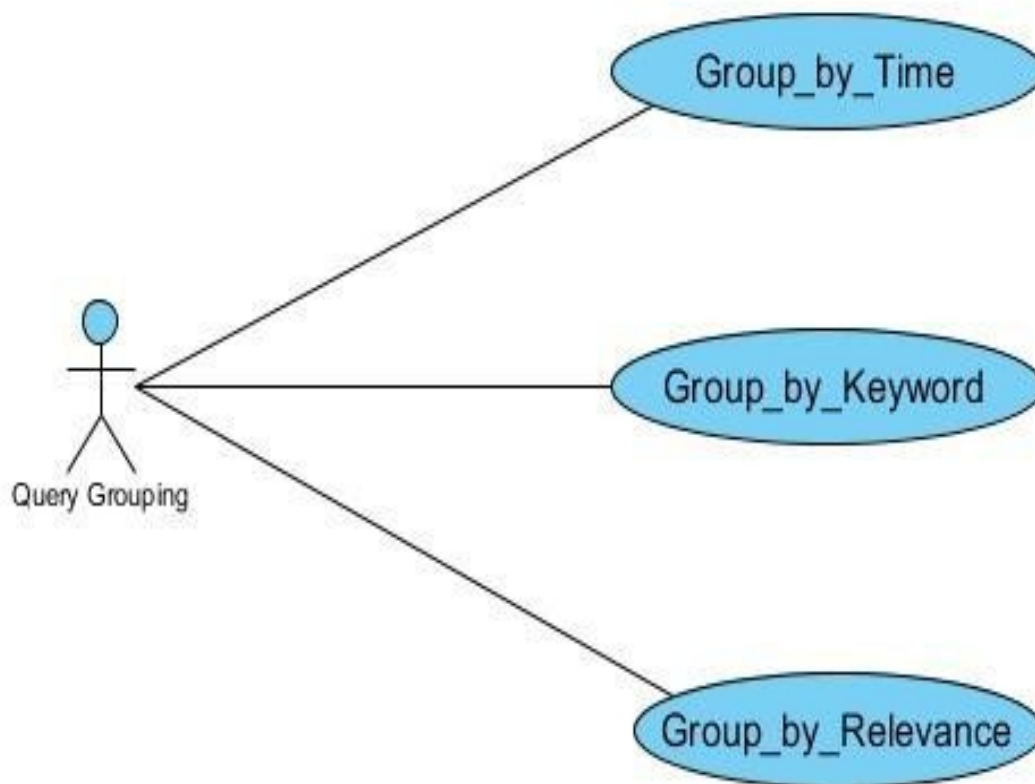
**Use Case Diagram 2:**

Fig.4.3 Use case Diagram



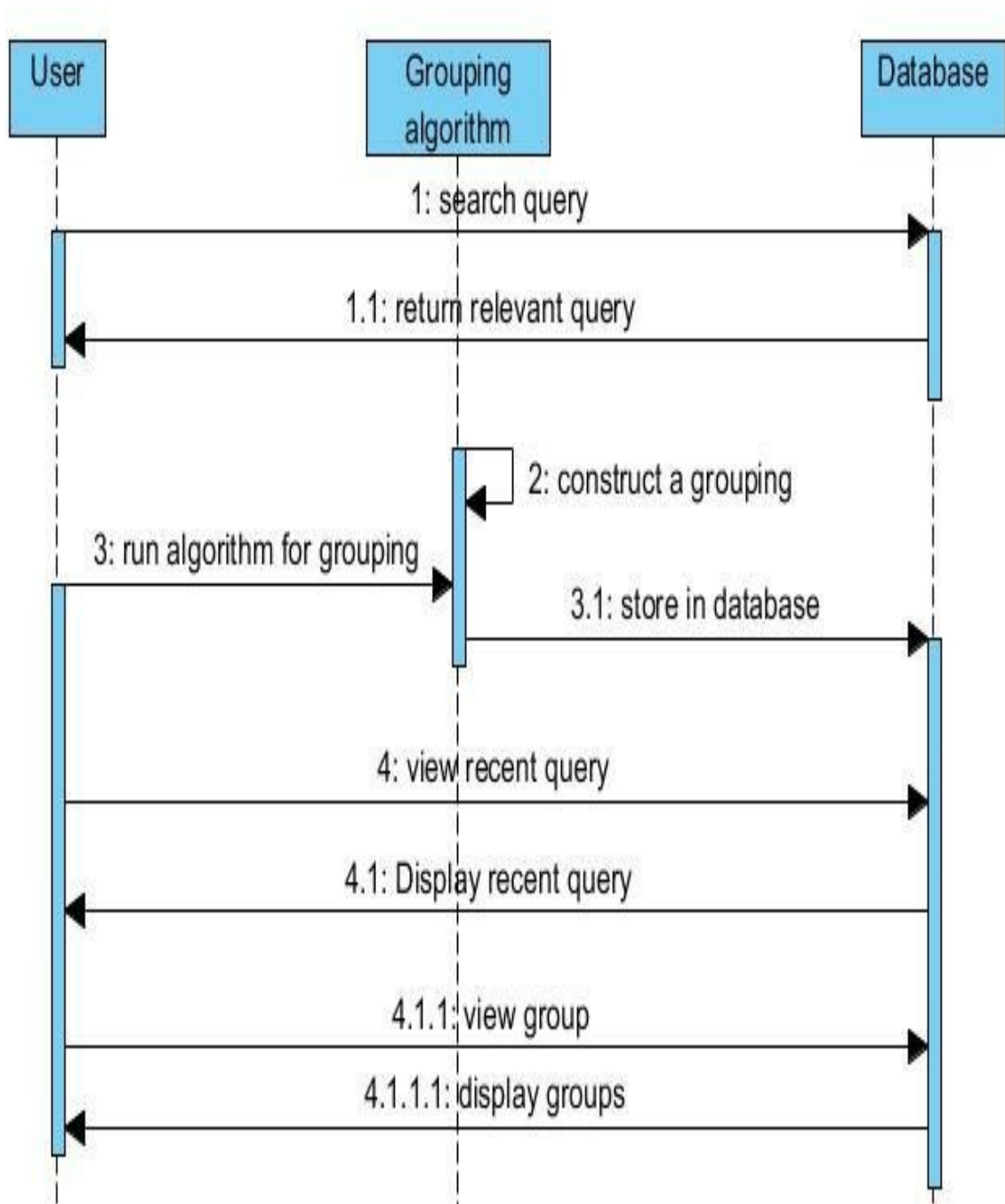
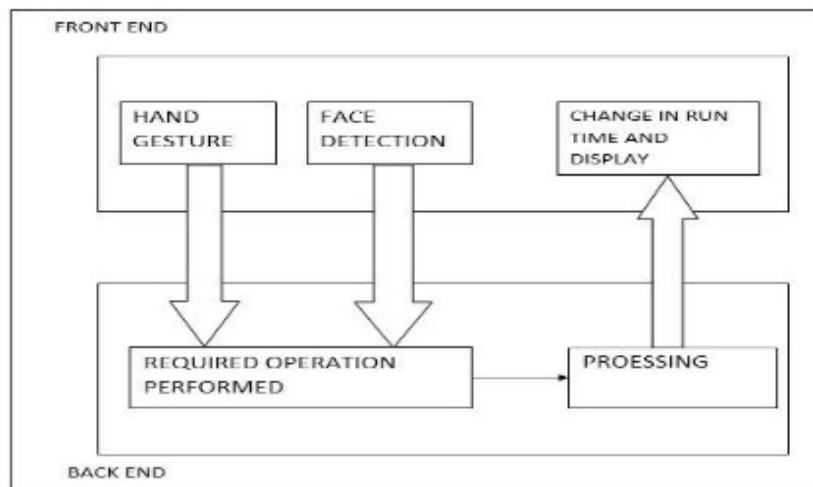
**Sequence Diagram:**

Fig.4.4. Sequence Diagram

## **5. IMPLEMENTATION**

## IMPLEMENTATION

Here as soon as we start the media player the media player will be opened with face detection mode i.e. it will start detecting human face and if a face is detected the video will keep on playing if there is no face the video will be paused. This media player has an option to run media player with either face detection mode or hand gesture mode and it can switch between them in run time. In hand gesture mode we control the mouse cursor to operate the media player.



### Module Description:

1. Face Detection
2. Hand gesture
3. Media player

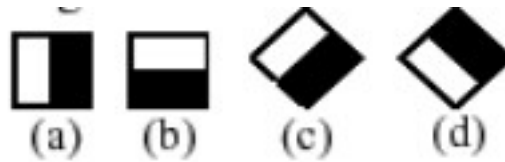
### 5.1 Face Detection:

OpenCV's face detector uses a method that Paul Viola and Michael Jones had published in 2001. Usually this approach detects the objects in images which combines four key concepts:

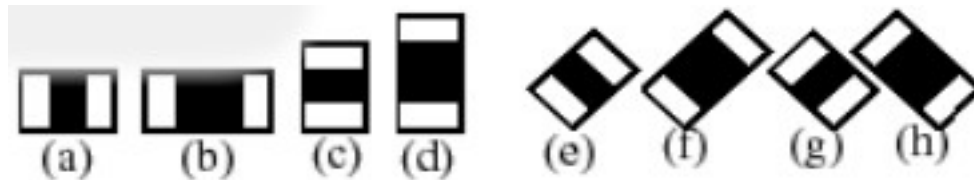
- Simple rectangular features which are called as Haar features.
- An integral Image for quick feature detection.
- A cascaded classifier to combine many features.

Following are the OpenCV's haar like features:-

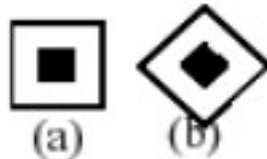
### 1. Edge features



### 2. Line features



### 3. Centre-Surround features



In this section the camera starts detecting a human face and till the time we are looking at the media player screen the video will be played and if we look somewhere else the video will be automatically paused. It will start playing the video as soon as it detects a face again.

## 5.2 Hand gesture:

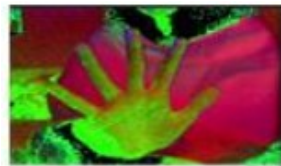
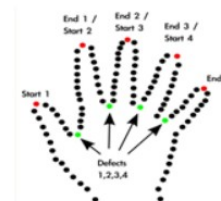
In this section the camera looks for the color bands. The user will wear 2 bands on his fingers they are green and blue color bands. Here the blue color band controls mouse cursor and green color band performs left click operation. Thus we can control the media player.

HSV (Hue, Saturation, Value) Color scheme:-

Used classical method to detect skin pixels By setting Upper & Lower bound values

$H_{min} \leq H \leq H_{max}$  {  $H_{min} 0$   $H_{max} 20 \dots \dots (1)$

$S_{min} \leq S \leq S_{max}$  {  $S_{min} 45$   $S_{max} 255 \dots \dots (2)$

**STEPS:-****1. INPUT IMAGE****2. HSV****3. BINARY IMAGE****4. CONVEX HULL****5. CONVEXITY DEFECTS****5.3 Media player:**

The media player is the important concept which will be used to play video and it will be controlled by functionalities like hand gesture and face detection.

## **6. TESTING**

## 6. TESTING

### **PURPOSE:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **6.1 Types of Tests**

#### **Unit testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Functional Testing:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

**Valid Input:** Identified classes of valid input must be accepted.

**Invalid Input:** Identified classes of invalid input must be rejected.

**Functions:** Identified functions must be exercised.

**Output:** Identified classes of application outputs must be exercised.

**Systems/Procedures:** Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.



## 6.2 Test Strategy and Approach

Field-testing will be performed manually and functional tests will be written in detail.

### Test Objectives:

- All field entries must work properly.
- Media player must be running state.
- The entry screen, messages and responses must not be delayed.

### Features to be tested:

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

## 6.3 Test Results

All the test cases mentioned above passed successfully. No defects encountered.

## **7 RESULTS AND PERFORMANCE EVALUATION**

## 7.1 RESULTS

Using the proposed system the following results can be observed.

### Result of Travelling Database:

```
C:\Program Files\Java\jdk1.6.0\bin>javac databasecon.java C:\
Program Files\Java\jdk1.6.0\bin>java databasecon
```

Colname	Datatype	Length	Decimals	Allow Null
Id	int	20	0	NOT NULL (primary key)
Category	varchar	50	0	NULL
S.cat	varchar	200	0	NULL
Keyword	varchar	225	0	NULL
Des	longtext	0	0	NULL
Currtime	varchar	30	0	NULL
Currrdate	varchar	20	0	NULL
Status	varchar	20	0	NULL

## 7.2 PERFORMANCE EVALUATION

While running the normal search engines there is no query grouping is done. But when we apply this relevance based searching algorithm query grouping is performed. User feels comfort using this query group searching.

Thus there was efficiency in searching.

## **8 CONCLUSION & FUTURE WORK**

## **8.1 CONCLUSION:**

The main concern of this project is to help the user get best experience of using a media player. We have tried to achieve this goal by automating the media player in a wide extent. We are doing this by using face recognition and hand gestures for controlling varied features of the media player such as pausing and starting the video again when the user isn't looking at the screen(for which face recognition is used), and controlling functions as forwarding , back-warding, volume up and volume down(for which hand gestures are used).

## **8.2 FUTURE WORK:**

This project can be more interactive with the help of tracking real time hand movements and controlling mouse pointer on screen. The short coming of requiring a plain background can be overcome with the help of Background Image Subtractions or Machine Learning Techniques.

The performance of the proposed method highly depends on the result of hand detection. If there are moving objects with the color similar to that of the skin, the objects exist in the result of the hand detection and then degrade the performance of the hand gesture recognition.

## **9 APPENDIX**

## Introduction to UML:

### UML Approach:

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built.

Definition: UML is a general purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system.

**UML is a language:** It provides vocabulary and rules for Communications. These vocabulary and rules focus on conceptual and physical representation of a system. So, UML is standard language for software blueprint.

**UML is a language for visualization:** The UML more than just a bunch of graphical symbols. Rather behind each symbol in UML notation is a well defined semantics. In this manner, one we can write a model in UML, and other tools can also interpret that model.

**UML is a language for specifying:** Specifying means building models that are precise, unambiguous and complete. In particular, the UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

**UML is language for constructing:** UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

**UML is a language for Documenting:** UML provides variety of documents in addition raw executable codes. These artifacts include:

- Requirements
- Architecture
- Design
- Source Code
- Project Plans
- Tests
- Prototypes

### **Goal of UML:**

The primary goal in the design of the UML is:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.



**Uses of UML:**

The UML is intended primarily for software intensive systems. It has been used effectively for such domain as

1. Enterprise information system
2. Banking and financial services
3. Telecommunications
4. Transportations
5. Defense/Aerospace
6. Retails
7. Medical Electronics
8. Scientific Fields
9. Distributed Web

**Rules of UML:**

The UML has semantic rules for

- **Names:** it will call things, relationship and diagrams.
- **Scope:** the content that gives specific meaning to a name.
- **Visibility:** how those names can be seen and used by others.
- **Integrity:** how things properly and consistently relate to another.
- **Execution:** what it means is to run or simulate a dynamic model.

**Building blocks of UML:**

The vocabulary of the UML encompasses 3 kinds of building blocks:

1. Things
2. Relationships
3. Diagrams

**1. Things:** things are the data abstractions that are first class citizens in a model. Things are of 4 types

- Structural Things
- Behavioral Things
- Grouping Things
- An notational things

**2. Relationships:** Relationships tie the things together.

Relationships in the UML are

- Dependency
- Association
- Generalization
- Specialization

**3. Diagrams:** Diagrams in the UML are of 2 types

- Static Diagrams
- Dynamic Diagrams
  
- Class Diagram
- Object Diagram
- Component Diagram
- Deployment Diagram

Dynamic Diagrams consists of

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- State chart Diagram
- Activity Diagram

## 9.2 SOFTWARE ENVIRONMENT

### OVERVIEW OF PYTHON:

Python is a general-purpose, object-oriented programming language with high-level programming capabilities. It has become famous because of its apparent and easily understandable syntax, portability, and easy to learn.

Python's flexibility is why the first step in every Python project must be to think about the project's audience and the corresponding environment where the project will run. It might seem strange to think about packaging before writing code, but this process does wonders for avoiding future headaches.

Python is a programming language that includes features of C and Java. It provides the style of writing an elegant code like C, and for object-oriented programming, it offers classes and objects like Java. Python is used to create web and desktop applications, and some of the most popular web applications like Instagram, YouTube, Spotify all have been developed in Python. You can also develop the next big thing by using Python.

For a long time many operating systems, including Mac and Windows, lacked built-in package management. Only recently did these OSes gain so-called "app stores", but even those focus on consumer applications and offer little for developers.

Computing as we know it is defined by the ability to execute programs. Every operating system natively supports one or more formats of program they can natively execute.

**Features:**

Python is gaining good popularity in the programming community; there are many reasons behind this.

- **Interpreted Language:** Python is processed at run time by Python Interpreter.
- **Object-Oriented Language:** It supports object-oriented features and techniques of programming.
- **Interactive Programming Language:** Users can interact with the python interpreter directly for writing programs.
- **Easy language:** Python is easy to learn, especially for beginners.
- **Straightforward Syntax:** The formation of python syntax is simple and straightforward, which also makes it popular.
- **Easy to read:** Python source-code is clearly defined and visible to the eyes.
- **Portable:** Python codes can be run on a wide variety of hardware platforms having the same interface.
- **Extendable:** Users can add low level-modules to Python interpreter.
- **Scalable:** Python provides an improved structure for supporting large programs then shell-scripts.

**Advantages:**

The diverse application of the Python language is a result of the combination of features which give this language an edge over others. Some of the benefits of programming in Python include:

1. **Presence of Third Party Modules**
2. **Extensive Support Libraries**
3. **Open Source and Community Development**
4. **Learning Ease and Support Available**
5. **User-friendly Data Structures**
6. **Productivity and Speed**

## OVERVIEW OF OPENCV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it is integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

**Applications of OpenCV:** There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anomaly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

**OpenCV Functionality:**

- Image/video I/O, processing, display (core, img proc, high gui)
- Object/feature detection (obj detect, features2d, non free)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videos tab)
- Computational photography (photo, video, super res)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

**Image-Processing:**

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.

If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

**Digital-Image :**

An image may be defined as a two-dimensional function  $f(x, y)$ , where  $x$  and  $y$  are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates  $(x, y)$  is called the intensity or grey level of the image at that point.

In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function  $f(x, y)$  at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.

Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

**Image processing basically includes the following three steps:**

1. Importing the image
2. Analysing and manipulating the image
3. Output in which result can be altered image or report that is based on image analysis

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

## 9.3 CODING

### **Main.py**

```

import math
from tkinter import font as tkFont
import numpy as np
import cv2 # required 3+
import tkinter as tk
from threading import Thread
import queue as Queue
import time
import os

from commands import *

# Frontal face classifier is imported here
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

request_queue = Queue.Queue()
result_queue = Queue.Queue()
t = None
debug = False
enable_commands = False
REALLY_NOT_DEBUG = True
CHANGE_VOLUME = False
COOLDOWN = 5
LAST_TIME = time.time()

def submit_to_tkinter(cb, *args, **kwargs):
    request_queue.put((cb, args, kwargs))
    return result_queue.get()

def debug_toggle():
    global debug
    debug = not debug

def toggle_commands():
    global enable_commands
    enable_commands = not enable_commands

```



```

def main_tk_thread():
    global t

    def timertick():
        try:
            cb, args, kwargs = request_queue.get_nowait()
        except Queue.Empty:
            pass
        else: # if no exception was raised
            retval = cb(*args, **kwargs)
            result_queue.put(retval)
            # reschedule after some time
            t.after(10, timertick)

    # create main Tk window
    t = tk.Tk()
    t.title("Debug controls")
    t.geometry('%dx%d+%d+%d' % (320, 320, 850, 200))
    # set font for labels
    font = tkFont.Font(family="Arial", size=18, weight=tkFont.BOLD)
    # create buttons, labels
    tc = tk.Button(text='enable commands', name='ec',
command=toggle_commands, width='15')
    tc.place(x=20, y=210)
    b = tk.Button(text='debug mode', name='dbg', command=debug_toggle,
width='15')
    b.place(x=20, y=260)
    hull = tk.Label(t, name="hull", text="None", font=font)
    hull.place(x=20, y=10)
    defects = tk.Label(t, name="defects", text="None", font=font)
    defects.place(x=20, y=60)
    defects_filtered = tk.Label(t, name="defects_filtered", text="None",
font=font)
    defects_filtered.place(x=20, y=110)
    command = tk.Label(t, name="command", text="None", font=font)
    command.place(x=20, y=160)
    en_command = tk.Label(t, name="en_command", text="None")
    en_command.place(x=160, y=215)
    en_dbg = tk.Label(t, name="en_dbg", text="None")
    en_dbg.place(x=160, y=265)
    # start timer a.k.a. scheduler
    timertick()
    # main Tk loop
    t.mainloop()

```

```

# setters for Tk GUI elements
def hull_label(a):
    t.children["hull"].configure(text=str("All hulls = %s " % a))

def defects_label(a):
    t.children["defects"].configure(text=str("All defects = %s" % a))

def defects_filtered_label(a):
    t.children["defects_filtered"].configure(text=str("Defects filtered = %s" %
a))

def command_label(a):
    t.children["command"].configure(text=str("Command = %s" % a))

def en_command_label(a):
    t.children["en_command"].configure(text=str("(%s)" % a))

def en_dbg_label(a):
    t.children["en_dbg"].configure(text=str("(%s)" % a))

def check_command(c, exe):
    if c == 0:
        return "NOACTION"
    elif c == 1:
        return "NOACTION"
    elif c == 2:
        if REALLY_NOT_DEBUG and exe:
            move_next()
        return "NEXT"
    elif c == 3:
        if REALLY_NOT_DEBUG and exe:
            move_prev()
        return "PREVIOUS"
    elif c == 4:
        if CHANGE_VOLUME and exe:
            vol_down()
        return "VOLUME CONTROL DOWN"
    elif c == 5:
        if CHANGE_VOLUME and exe:
            vol_down()
        return "VOLUME CONTROL UP"
    return None

```

```

if __name__ == '__main__':
    t = Thread(target=main_tk_thread)
    t.start()

    # Flag is used to pause and play the video [ if flag is 1 then the video
    plays else it doesn't ]
    Pauseflag = 0

    cap = cv2.VideoCapture(0)

    while cap.isOpened():
        # 1. Constructing Region Of Interest (ROI)
        # read frame from camera
        _, img = cap.read()
        # add rectangle for 'target scanning area' 300x300
        cv2.rectangle(img, (350, 350), (50, 50), (0, 255, 0), 0)
        # crop input image to 300x300
        crop_img = img[50:350, 50:350]
        # convert image to gray scale
        gray = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
        gray1 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        #face identification
        faces = face_cascade.detectMultiScale(gray1, 1.3, 5)
        if debug:
            cv2.imshow('Gray scale', gray)
        # use Gaussian blur
        blur = cv2.GaussianBlur(src=gray, ksize=(35, 35), sigmaX=0)
        if debug:
            cv2.imshow('Blurred', blur)
        # apply threshold to get black and white (binary) image (ROI)
        _, thresh1 = cv2.threshold(blur, 127, 255, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU)
        if debug:
            cv2.imshow('Threshold', thresh1)

        # 2. Analyze ROI
        # find contours
        contours, hierarchy = cv2.findContours(thresh1, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
        # contours, hierarchy = cv2.findContours(thresh1.copy(),
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

        # get max contour area
        cnt = max(contours, key=lambda x: cv2.contourArea(x))

```

```

# convex hull
x, y, w, h = cv2.boundingRect(cnt)
cv2.rectangle(crop_img, (x, y), (x + w, y + h), (0, 0, 255), 0)
hull = cv2.convexHull(cnt)

# contours
drawing = np.zeros(crop_img.shape, np.uint8)
cv2.drawContours(drawing, [cnt], 0, (0, 255, 0), 0)
cv2.drawContours(drawing, [hull], 0, (0, 0, 255), 0)
hull = cv2.convexHull(cnt, returnPoints=False)

# we find convexity defects, which is the deepest point of deviation on
the contour
defects = cv2.convexityDefects(cnt, hull)
count_defects = 0
cv2.drawContours(thresh1, contours, -1, (0, 255, 0), 3)
for i in range(defects.shape[0]):
    s, e, f, d = defects[i, 0]
    start = tuple(cnt[s][0])
    end = tuple(cnt[e][0])
    far = tuple(cnt[f][0])
    # dist
    a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
    b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
    c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
    angle = math.acos((b**2 + c**2 - a**2) / (2 * b * c)) * 57
    if angle <= 90:
        count_defects += 1
        cv2.circle(crop_img, far, 3, [255, 0, 0], -1)
        # dist = cv2.pointPolygonTest(cnt, far, True)
        cv2.line(crop_img, start, end, [0, 255, 0], 2)
        if debug:
            cv2.circle(crop_img, far, 5, [0, 0, 255], -1)

# show analyzed input/output
if debug:
    all_img = np.hstack((drawing, crop_img))
    cv2.imshow('Contours vs image', all_img)

cv2.imshow('Input', img)

k = cv2.waitKey(50)
# got ESC key? if yes - exit!
if k == 27:
    break

```

```

elif k == 99: # for 'c' toggle command execution
    print ('c input')
    toggle_commands()
elif k == 100: # for 'd' toggle debug mode
    print ('d input')
    debug_toggle()

# do not 'change' command to quickly and wait after last one
if time.time() - LAST_TIME > COOLDOWN and enable_commands:
    exe = True
    LAST_TIME = time.time()
else:
    exe = False

# check what command to execute and run it
com = check_command(count_defects, exe)

delta = time.time() - LAST_TIME
to_next = COOLDOWN - delta
if to_next < 0:
    to_next = 0

if REALLY_NOT_DEBUG and exe:
    for (x, y, w, h) in faces:
        play()
        Pauseflag = 1 # Face is detected hence play the video
continuesly
    if Pauseflag == 0:
        pause()
        Pauseflag = 0

# submit some data to GUI
submit_to_tkinter(hull_label, str(hull.shape[0]))
submit_to_tkinter(defects_label, str(defects.shape[0]))
submit_to_tkinter(defects_filtered_label, str(count_defects))
submit_to_tkinter(en_command_label, str("%s, cooldown %.2fs." %
(enable_commands, to_next)))
submit_to_tkinter(en_dbg_label, debug)
if com:
    submit_to_tkinter(command_label, com)

```

**Commands.py:**

```
from subprocess import check_call

__all__ = ['play', 'pause', 'move_prev', 'move_next', 'vol_up', 'vol_down']

def play():
    check_call(['vlc-ctrl', 'play'])

def pause():
    check_call(['vlc-ctrl', 'pause'])

def move_next():
    check_call(['vlc-ctrl', 'next'])

def move_prev():
    check_call(['vlc-ctrl', 'prev'])

def vol_up():
    check_call(['vlc-ctrl', 'volume', '+0.1'])

def vol_down():
    check_call(['vlc-ctrl', 'volume', '-0.1'])
```

## 9.4 List of Figures

<b>S. No</b>	<b>Name of the Figure</b>	<b>Page No</b>
4.1	Class Diagram	15
4.2	Usecase Diagram	16
4.3	Usecase Diagram	17
4.4	Sequence Diagram	18
5.1	Query Grouping	20
5.2	Search History	21
9.1	Home Page	69
9.2	Input Query	70
9.3	Query Result	71
9.4	Public History	72
9.5	Registration Page	73
9.6	User Login Page	74
9.7	User Home Page	75
9.8	User Details Page	76
9.9	User Search History by Date	77
9.10	Database Login Page	78
9.11	Database Home Page	79
9.12	User Details	80
9.13	Add Travelling Details	81
9.14	View History Graph	82
9.15	Password Update	83
9.16	Query Groups	84

## 9.5 List of Tables

<b>S. No</b>	<b>Name of the table</b>	<b>Page No.</b>
9.1.1	Travelling Database Structure	85
9.1.2	Travelling Database	86
9.1.3	Registration Database Structure	87
9.1.4	Registration Database	87
9.1.5	User History Database Structure	88
9.1.6	User History Database	88
9.1.7	Public History Database Structure	89
9.1.8	Public History Database	89



## 9.6 LIST OF ABBREVIATIONS

- UML            UNIFIED MODELING LANGUAGE
- SQL            STRUCTURED QUERY LANGUAGE
- DBMS          DATA BASE MANAGEMENT SYSTEM
- JDBC          JAVA DATABASE CONNECTIVITY
- HTML          HYPER TEXT MARKUP LANGUAGE
- CSS            CASCADING STYLE SHEETS
- RDBMS        RELATIONAL DATABASE MANAGEMENT SYSTEM

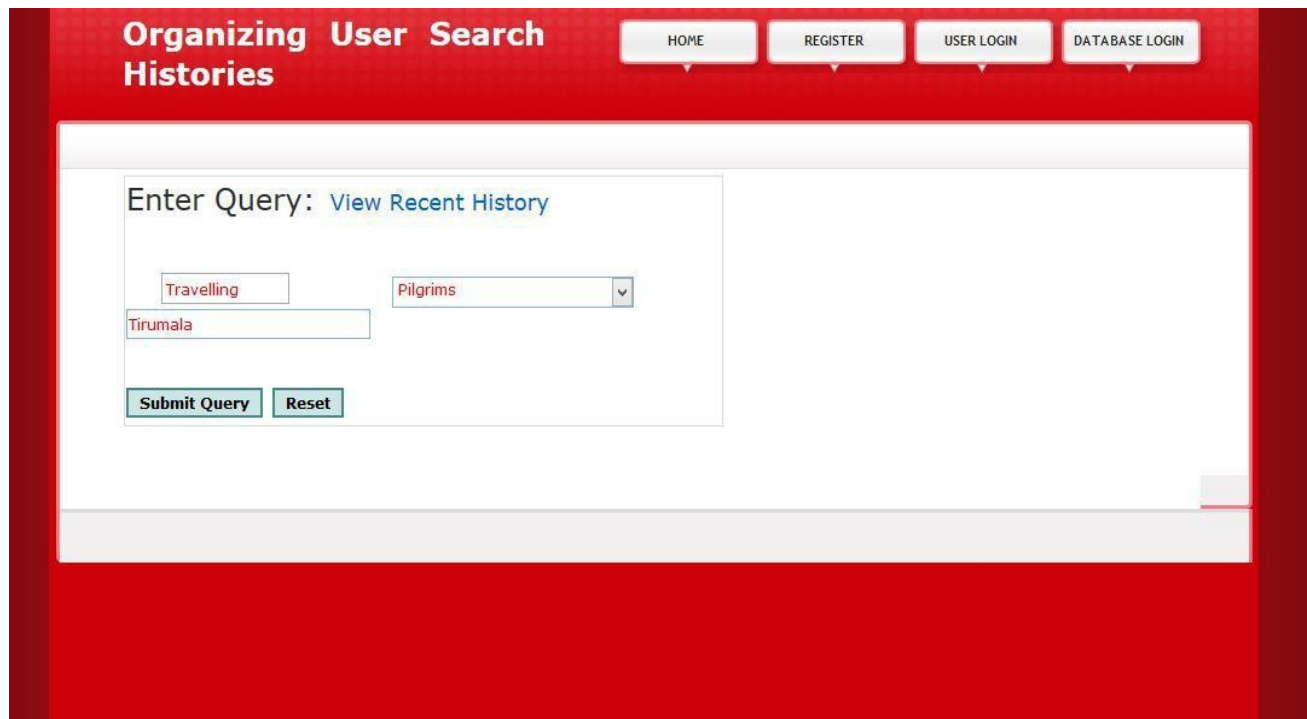
## 9.7 SCREENSHOTS

### Homepage:

The screenshot shows a web application interface with a red header. The header contains the title 'Organizing User Search Histories' on the left and four navigation buttons ('HOME', 'REGISTER', 'USER LOGIN', 'DATABASE LOGIN') on the right. Below the header is a white content area. Inside this area is a search form with the label 'Enter Query: View Recent History'. The form includes a text input field containing the word 'Travelling', a dropdown menu with the text '-----Select-----', and two buttons labeled 'Submit Query' and 'Reset'.

Fig. 9.1 Home page

## Home Page-Input Query:



The screenshot shows a web application interface with a red header. The header contains the text "Organizing User Search Histories" on the left and four navigation buttons: "HOME", "REGISTER", "USER LOGIN", and "DATABASE LOGIN" on the right. Below the header is a white search form. The form has a label "Enter Query:" followed by a blue link "View Recent History". There are three input fields: a text field containing "Travelling", a dropdown menu showing "Pilgrims" with a downward arrow, and a text field containing "Tirumala". At the bottom of the form are two buttons: "Submit Query" and "Reset".

Fig. 9.2 Input Query

## Home Page-Output Results:

The screenshot displays a web application interface with a red header. The header contains the text "Organizing User Search Histories" and four navigation buttons: "HOME", "REGISTER", "USER LOGIN", and "DATABASE LOGIN". Below the header, the main content area is divided into two columns. The left column, titled "View Recent History", contains a "Related Search" section with checkboxes for "Travelling" and "Pilgrims". Below this, the "Title: Tirumala" is displayed, followed by a detailed "Description" of Tirumala as a hill town in the Chittoor district of Andhra Pradesh, India. The right column, titled "Related Results", lists several related search results: "Kapila theertham", "Kapila theertham", "Kapila theertham", "Kanipakam", "Kanchipuram", "Sri Kalahasthi", "Yadaqirigutta", "Kanakadurgama Temple", and "Ayyappa Temple".

**Organizing User Search Histories**

HOME REGISTER USER LOGIN DATABASE LOGIN

**View Recent History**

Related Search

☐ Travelling

☐ Pilgrims

Title: **Tirumala**

Description: Tirumala is a hill town in the Chittoor district of Andhra Pradesh, India. The town is a pilgrimage center for Hindus, with its Sri Venkateswara Temple. Tirumala is dedicated to the Shrimann Narayana, or Maha Vishnu. Tirumala is situated 26 kilometres (16 mi) north of Tirupati, 86 kilometres (53 mi) north-east of Chittoor, the district headquarters and 573 kilometres (356 mi) south-east of Hyderabad, the state capital. Tirumala Hill is 3,200 feet (980 m) above sea level, and the town covers an area of approximately 10.33 square miles (26.8 km<sup>2</sup>). In the area are seven peaks from the Seshachalam range, part of Eastern Ghats. The seven peaks represent the hood of Adisesha and are known as Seshadri, Neeladri, Garudadri, Anjanadri, Vrushabadri, Narayanadri and Venkatadri. The temple of Sri Venkateswara is on the seventh peak (Venkatadri). "Tirumala" is the combination of two words: Tiru means 'Holy' or 'Sacred' and mala means hills/mountain in Tamil Language. Therefore it

**Related Results**

Other Related Results

[Kapila theertham](#)

[Kapila theertham](#)

[Kapila theertham](#)

[Kanipakam](#)

[Kanchipuram](#)

[Sri Kalahasthi](#)

[Yadaqirigutta](#)

[Kanakadurgama Temple](#)

[Ayyappa Temple](#)

Fig. 9.3 Query Result

## Home Page -View Public History:

The screenshot displays a web application interface with a red header bar. The header contains the text "Organizing User Search Histories" on the left and four navigation buttons: "HOME", "REGISTER", "USER LOGIN", and "DATABASE LOGIN" on the right. Below the header, there is a white content area. At the top of this area, there is a form with a text input field labeled "Enter Date :" and a "Submit" button. Below this form, the section "Your Recent History" is displayed. It contains two entries, each with a checkbox, a status indicator, and a title. The first entry has a checked checkbox, a green checkmark, and the title "marriage". The second entry has an unchecked checkbox, a green checkmark, and the title "in". Both entries show a time of "19:17:58" and a date of "2013:3:2".

Organizing User Search Histories

HOME REGISTER USER LOGIN DATABASE LOGIN

Enter Date :  Submit

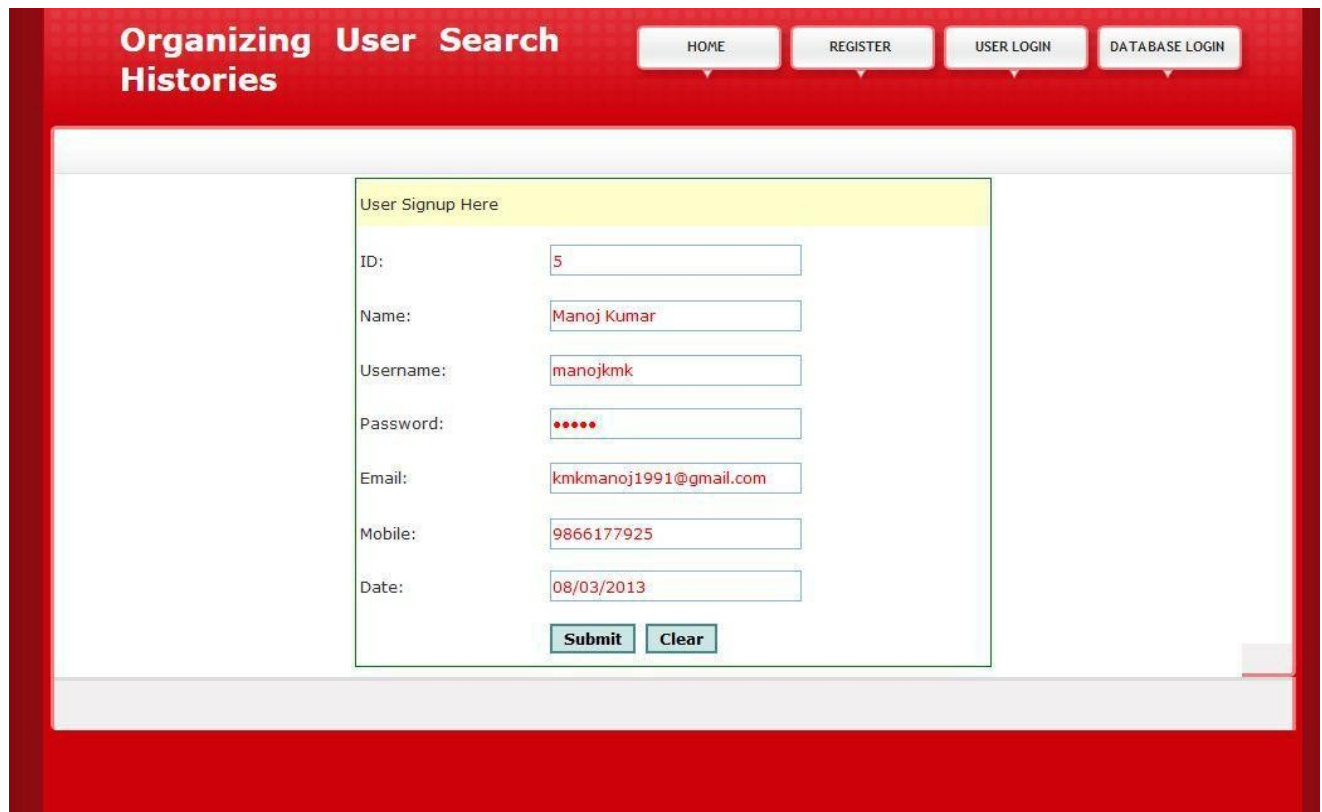
Your Recent History

☒ Travelling ✓  
☐ Marriage  
Title: marriage  
Time: 19:17:58  
Date: 2013:3:2

☐ Travelling ✓  
☐ Marriage  
Title: in  
Time: 19:17:58  
Date: 2013:3:2

Fig. 9.4 Public History

## User Registration Page:

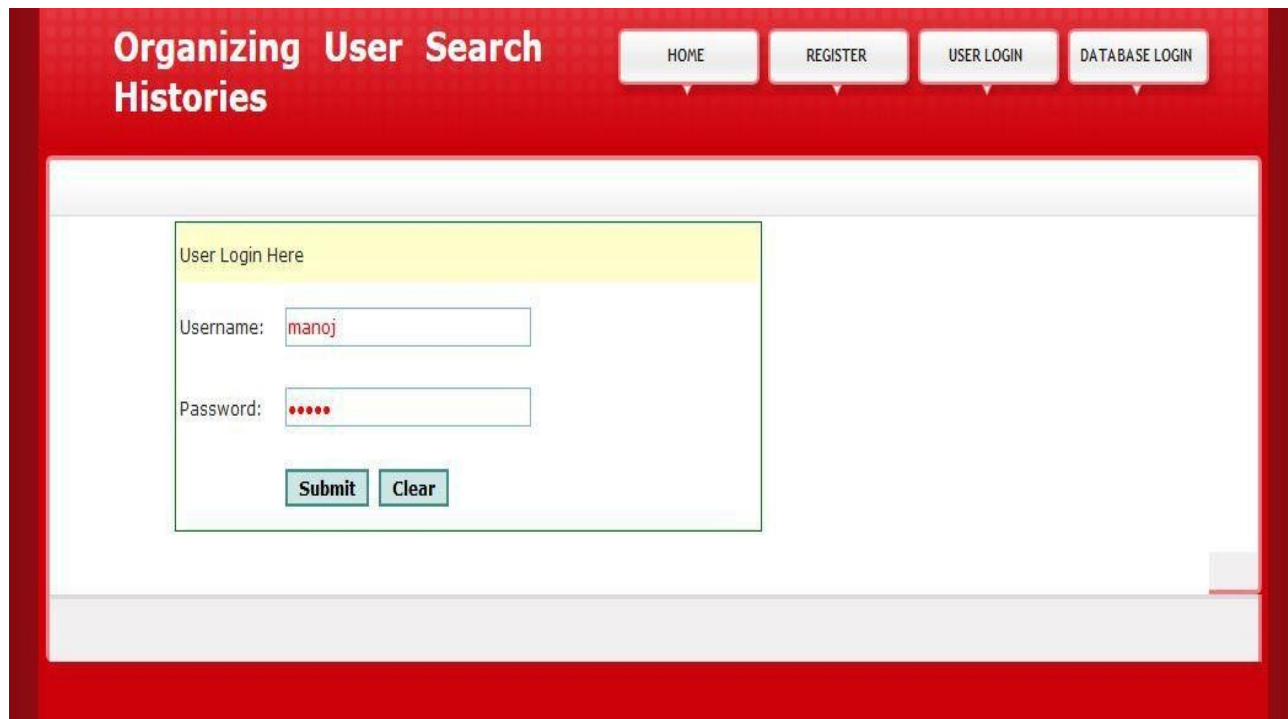


The screenshot displays a web application interface with a red header. The header contains the text "Organizing User Search Histories" on the left and four navigation buttons: "HOME", "REGISTER", "USER LOGIN", and "DATABASE LOGIN" on the right. Below the header, a white box contains a "User Signup Here" section. This section includes a form with the following fields: "ID:" (value: 5), "Name:" (value: Manoj Kumar), "Username:" (value: manojkmk), "Password:" (value: masked with red dots), "Email:" (value: kmkmanoj1991@gmail.com), "Mobile:" (value: 9866177925), and "Date:" (value: 08/03/2013). At the bottom of the form are "Submit" and "Clear" buttons.

User Signup Here	
ID:	5
Name:	Manoj Kumar
Username:	manojkmk
Password:	.....
Email:	kmkmanoj1991@gmail.com
Mobile:	9866177925
Date:	08/03/2013
<input type="button" value="Submit"/> <input type="button" value="Clear"/>	

Fig. 9.5 Registration Page

## User Login Page:



The screenshot displays a web application interface with a red header. On the left side of the header, the text "Organizing User Search Histories" is visible. On the right side, there are four navigation buttons: "HOME", "REGISTER", "USER LOGIN", and "DATABASE LOGIN". The "USER LOGIN" button is highlighted. Below the header, there is a white rectangular area containing a login form. The form has a yellow header section labeled "User Login Here". Below this, there are two input fields: "Username:" with the text "manoj" and "Password:" with five red dots. At the bottom of the form, there are two buttons: "Submit" and "Clear".

Fig. 9.6 User Login Page

## User Home Page:

The screenshot displays a web application interface with a red header bar. On the left side of the header, the text "Organizing User Search Histories" is written in white. On the right side, there are four white buttons with black text: "USER HOME", "YOUR DETAILS", "RECENT HISTORY", and "LOGOUT". Below the header, the main content area has a white background. It begins with the label "Enter Query:". Below this label, there is a text input field containing the word "Travelling", followed by a dropdown menu showing "-----Select-----" with a downward arrow. Underneath these elements is another empty text input field. At the bottom of this section, there are two buttons: "Submit Query" and "Reset".

Fig. 9.7 User Home Page



## User Update Details:

The screenshot displays a web application interface with a red header. The header contains the text 'Organizing User Search Histories' on the left and four navigation buttons: 'USER HOME', 'YOUR DETAILS', 'RECENT HISTORY', and 'LOGOUT'. The main content area is white and features a form titled 'Update Password Here' in a grey box. The form contains several input fields with red text labels and values: ID (1), Name (Manoj Kumar), Username (manoj), Password (manoj), Email (kmkmanoj1991@gmail.com), Mobile (9866177925), and Date (25/02/2013). An 'Update' button is located at the bottom of the form.

Update Password Here	
ID:	1
Name:	Manoj Kumar
Username:	manoj
Password:	manoj
Email:	kmkmanoj1991@gmail.com
Mobile:	9866177925
Date:	25/02/2013
<input type="button" value="Update"/>	

Fig. 9.8 User Details Page

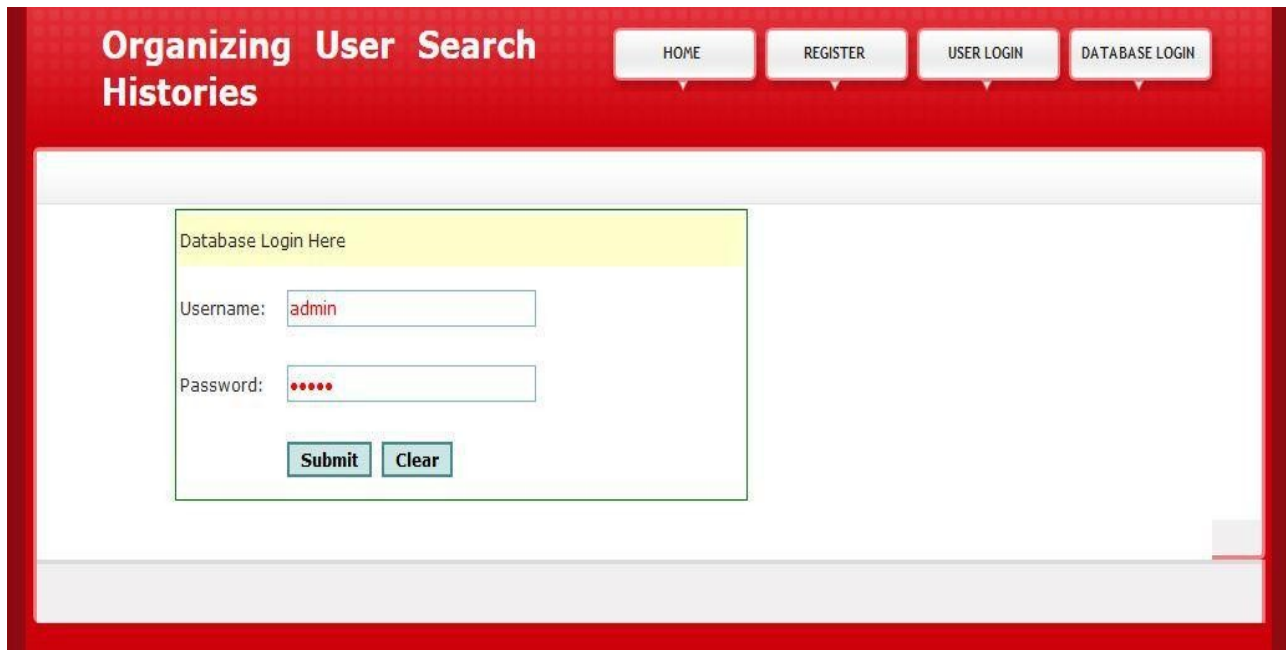
## User Search History:

The screenshot shows a web application titled "Organizing User Search Histories". At the top, there is a red navigation bar with four buttons: "USER HOME", "YOUR DETAILS", "RECENT HISTORY", and "LOGOUT". Below the navigation bar, there is a search area with a text input field labeled "Enter Date :" containing the value "2013:3:4" and a "Submit" button. Below the search area, there is a section titled "Your Recent History" which displays two search results. Each result includes a checkbox, a category name, a title, a time, and a date.

Category	Title	Time	Date
<input type="checkbox"/> Travelling ✓ <input type="checkbox"/> Pilgrims	Tirupati	20:11:22	2013:3:4
<input type="checkbox"/> Travelling ✓ <input type="checkbox"/> Hotels	bhimas	20:12:14	2013:3:4

Fig.9.9 User Search History by Date

## Database Login Page:



The screenshot shows a web application interface with a red header. On the left, the text "Organizing User Search Histories" is displayed. On the right, there are four navigation buttons: "HOME", "REGISTER", "USER LOGIN", and "DATABASE LOGIN". The "DATABASE LOGIN" button is highlighted. Below the header, there is a white content area. Inside this area, there is a yellow box titled "Database Login Here". Within this box, there are two input fields: "Username:" with the value "admin" and "Password:" with five red dots. Below these fields are two buttons: "Submit" and "Clear".

Fig.9.10 Database Login Page

## Database Homepage:

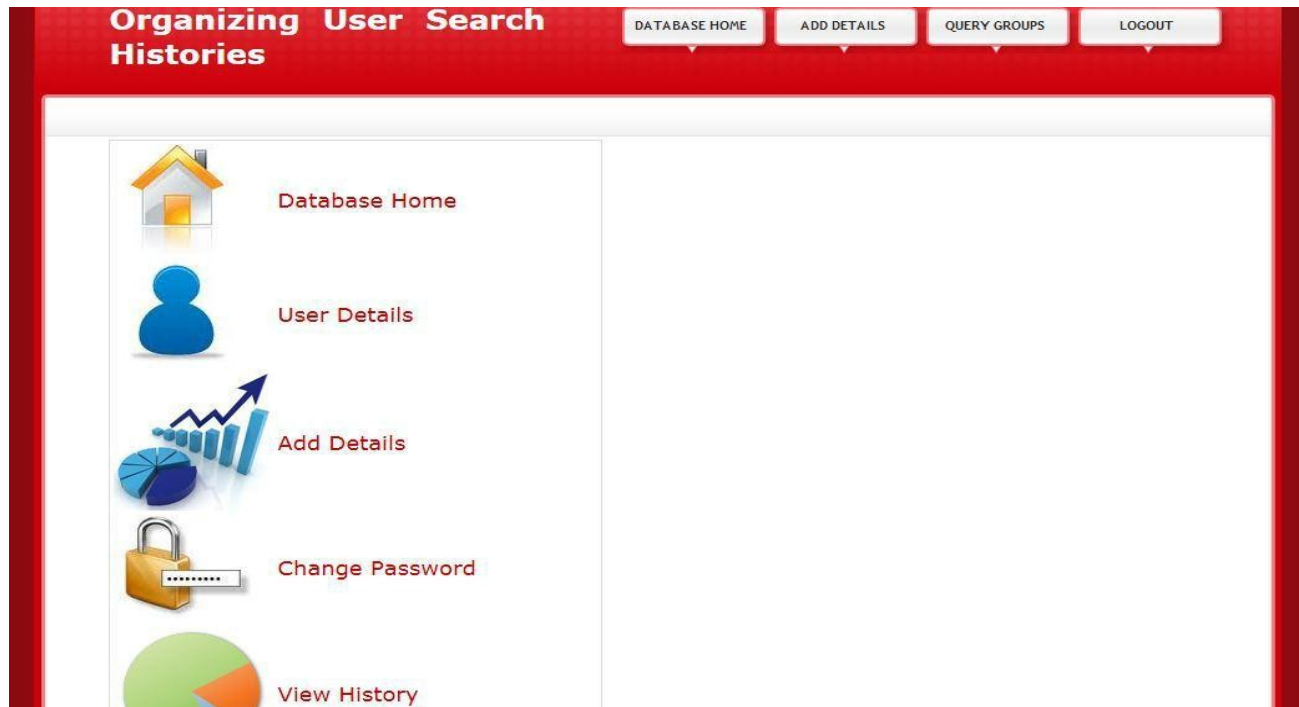
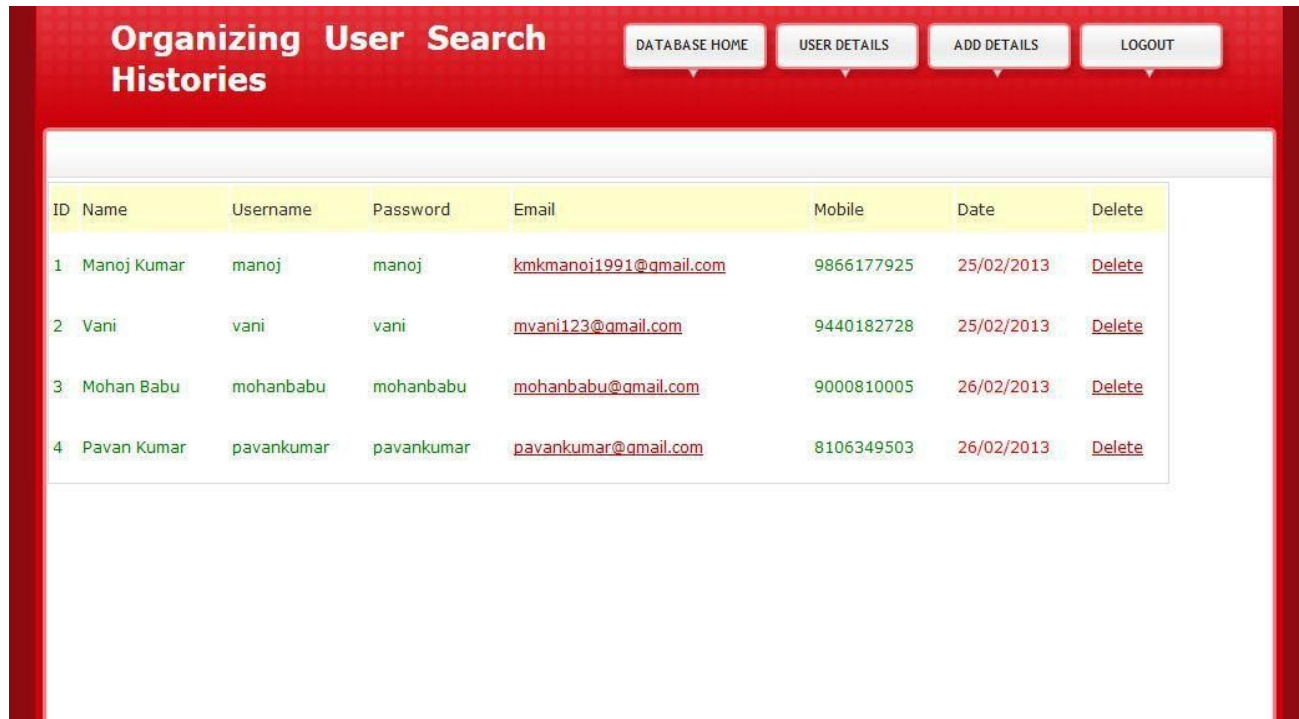


Fig. 9.11 Database Home Page

## Database Page - User Details:



ID	Name	Username	Password	Email	Mobile	Date	Delete
1	Manoj Kumar	manoj	manoj	<a href="mailto:kmkmanoj1991@gmail.com">kmkmanoj1991@gmail.com</a>	9866177925	25/02/2013	<a href="#">Delete</a>
2	Vani	vani	vani	<a href="mailto:mvani123@gmail.com">mvani123@gmail.com</a>	9440182728	25/02/2013	<a href="#">Delete</a>
3	Mohan Babu	mohanbabu	mohanbabu	<a href="mailto:mohanbabu@gmail.com">mohanbabu@gmail.com</a>	9000810005	26/02/2013	<a href="#">Delete</a>
4	Pavan Kumar	pavankumar	pavankumar	<a href="mailto:pavankumar@gmail.com">pavankumar@gmail.com</a>	8106349503	26/02/2013	<a href="#">Delete</a>

Fig. 9.12 User Details

## Database Page - Add Travelling Details:

The screenshot displays a web application interface with a red header bar. On the left, the text 'Organizing User Search Histories' is shown. On the right, there are four buttons: 'DATABASE HOME', 'USER DETAILS', 'ADD DETAILS', and 'LOGOUT'. Below the header, a white box contains a form titled 'Add Details Here'. The form has the following fields: 'ID:' with the value '107', 'Category:' with the value 'Travelling', 'Sub Category:' with a dropdown menu showing '-----Select-----', 'Title:' with an empty text box, and 'Description:' with a larger empty text area. At the bottom of the form are two buttons: 'Submit' and 'Clear'.

Fig. 9.13 Add Travelling Details

## Database Page - View History Graph:

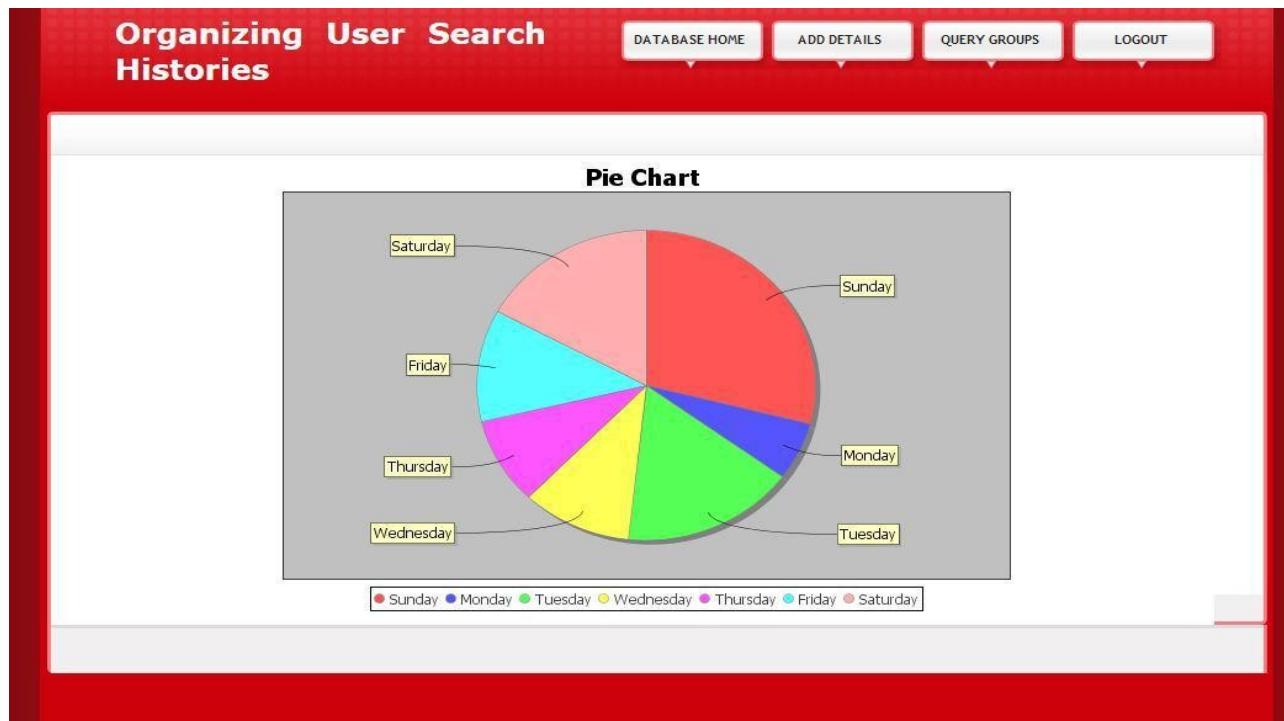
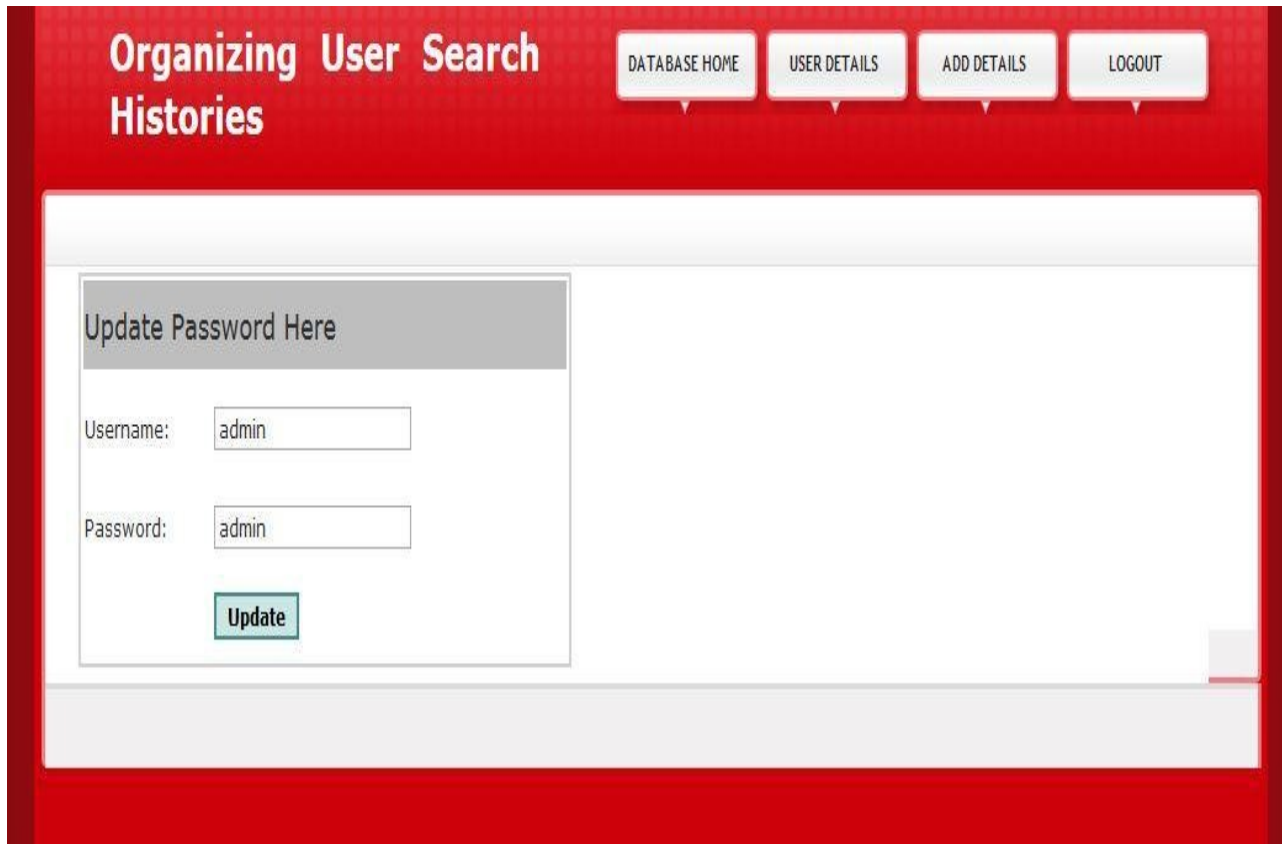


Fig.9.14 View History Graph

## Database Password Update:

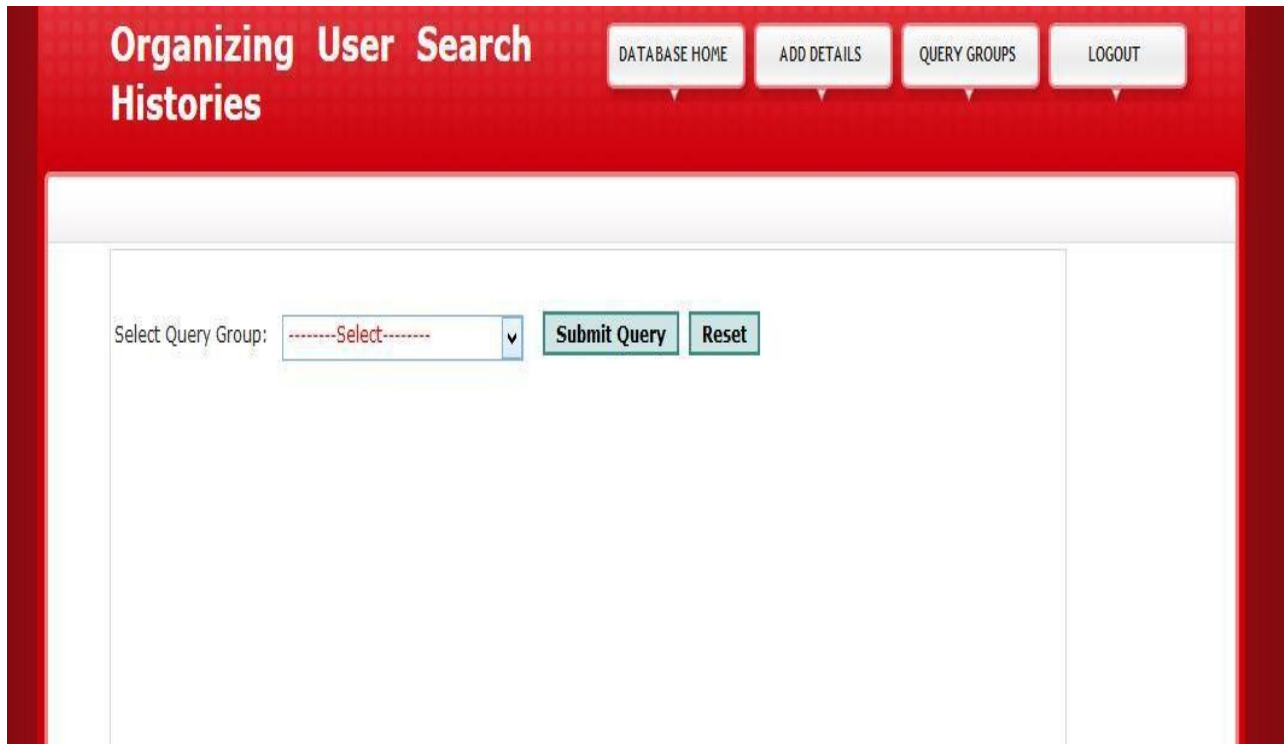


The screenshot displays a web application interface with a red header. On the left, the text 'Organizing User Search Histories' is shown. On the right, there are four buttons: 'DATABASE HOME', 'USER DETAILS', 'ADD DETAILS', and 'LOGOUT'. Below the header, a white box contains a form titled 'Update Password Here'. The form has two input fields: 'Username:' with the value 'admin' and 'Password:' with the value 'admin'. Below these fields is a green 'Update' button.

Fig. 9.15 Password Update



## Database Query Groups:



The screenshot displays a web application interface with a red header bar. On the left side of the header, the text "Organizing User Search Histories" is written in white. On the right side of the header, there are four white buttons with black text: "DATABASE HOME", "ADD DETAILS", "QUERY GROUPS", and "LOGOUT". Below the header, the main content area is white. It features a form with the label "Select Query Group:" followed by a dropdown menu showing "-----Select-----" with a downward arrow. To the right of the dropdown are two buttons: "Submit Query" and "Reset".

Fig. 9.16 Query Groups

## Travelling Database Structure:

[Table Design] financial @history (localhost)

File Edit Window

New Save Save As Add Field Insert Field Delete Field Primary Key Move Up Move Down

Fields Indexes Foreign Keys Triggers Options Comment

Name	Type	Length	Decimals	Allow Null	
uid	int	20	0	<input type="checkbox"/>	
cat	varchar	50	0	<input checked="" type="checkbox"/>	
scat	varchar	100	0	<input checked="" type="checkbox"/>	
tit	varchar	225	0	<input checked="" type="checkbox"/>	
des	longtext	0	0	<input checked="" type="checkbox"/>	
currenttime	varchar	30	0	<input checked="" type="checkbox"/>	
currentdate	varchar	20	0	<input checked="" type="checkbox"/>	
status	varchar	20	0	<input checked="" type="checkbox"/>	

Table 9.1.1 Travelling Database Structure

## Travelling Database:

(Table) financial @history (localhost)

File Edit View Window

Memo Hex Image Sort Ascending Sort Descending Remove Sort

uid	cat	scat	tit	des	currenttime	currentdate	status
1	Travelling	Pilgrims	Tirumala	Tirumala is a hill town in the Chittoor district of Andhra Pradesh, In	18:34:4	2013:3:2	Yes
2	Travelling	Job Searching	Job Searching Sites	Top Job Search Sites in India	19:3:57	2013:3:2	Yes
3	Travelling	Sightseeing	Ooty	Udhagamandalam , sometimes abbreviated to Udhagai and better	19:8:36	2013:3:2	Yes
4	Travelling	Marriage	Marriage Matrimony	BharatMatrimony - The No. 1 & Most Successful Indian Matrimonial	19:11:54	2013:3:2	Yes
5	Travelling	Colleges	Top Engineering Colleges	Top 25 Engineering Colleges in India	19:14:45	2013:3:2	Yes
6	Travelling	Hotels	Online Hotels Booking	It's these types of assurances which have helped Expedia become	19:31:24	2013:3:2	Yes
7	Travelling	Pilgrims	kapila theertham	The Holy Temple of Kapila Theertham is the only temple dedicated	18:37:30	2013:3:3	Yes
8	Travelling	Pilgrims	Kapila theertham	This huge temple stands at the entrance to a mountain cave in one	18:38:52	2013:3:3	Yes
9	Travelling	Pilgrims	Kapila theertham	In front of Kapila Lingam lies a very sacred Sarovara (lake), which	18:39:16	2013:3:3	Yes
10	Travelling	Marriage	Sri Padmavathi Kalyanmandap	Sri Padmavathi Kalyana Mandapam , Tiruchanoor Road , Tirupathi.	18:40:1	2013:3:3	Yes
11	Travelling	Colleges	sri venkateswara university (s	Sri Venkateswara University, is a state university, which was estat	18:40:28	2013:3:3	Yes
12	Travelling	Colleges	sri venkateswara university (s	To view full details of svu go to the link <a href="http://www.svuuniversity.in">http://www.svuuniversity.in</a>	18:40:59	2013:3:3	Yes
13	Travelling	Colleges	sree vidyanikethan engineering	Sree Vidyanikethan Engineering College was established in 1996 by	18:41:20	2013:3:3	Yes
14	Travelling	Colleges	sree vidyanikethan engineering	The college is affiliated to Jawaharlal Nehru Technological Universit	18:41:47	2013:3:3	Yes
15	Travelling	Colleges	sree vidyanikethan engineering	The B.tech Programs offered are: Bio Technology (Bio.Tech) Ci En	18:42:7	2013:3:3	Yes
16	Travelling	Hotels	Udayee International Hotel	The Udayee international is all about the experience that goes bey	18:42:45	2013:3:3	Yes
17	Travelling	Hotels	Udayee International Hotel	Our hotel is a place that will redefine more things than one. Udaye	18:43:7	2013:3:3	Yes
18	Travelling	Hotels	Udayee International Hotel	our services: High Security System,Internet, Fax, Facility,Multi Ch	18:43:27	2013:3:3	Yes
19	Travelling	Hotels	Udayee International Hotel	Distance from udayee to-Air Port - 12Km ,Renigunta Railway Stat	18:43:40	2013:3:3	Yes
20	Travelling	Hotels	Fortune Kences	Overview:-Fortune Kences Hotel showcases modern architecture r	18:44:23	2013:3:3	Yes
21	Travelling	Sightseeing	Goa	Ancestral Goa museum- , Ancestral Goa museum, also known as Big	18:44:55	2013:3:3	Yes
22	Travelling	Sightseeing	Goa	chapora fort-At a distance of approximately 12 km from Mapusa lie	18:45:20	2013:3:3	Yes
23	Travelling	Sightseeing	Goa	anjuna beach-aggd laterite boulders welcome you to the land of i	18:45:58	2013:3:3	Yes
24	Travelling	Job Searching	Business development	Business Development Manager / Ass. Sales Manager, 2nd Mar 20	18:46:31	2013:3:3	Yes
25	Travelling	Job Searching	Sales Development	Sales Development Manager, 2nd Mar 2013 Bench Mark Hires Sale	18:47:9	2013:3:3	Yes
26	Travelling	Job Searching	Sales Manager	Sales Manager, Relationship Manager, 1st Mar 2013 I HRCS Staff	18:47:52	2013:3:3	Yes
27	Travelling	Job Searching	Senior PHP Programmers	Senior PHP programmers required, 26th Feb 2013 Confidential Tiru	18:48:28	2013:3:3	Yes
28	Travelling	Job Searching	Engineer-Erection	Engineer-Erection/Fabrication/Site/Maintenance/Estimation/Design	18:48:54	2013:3:3	Yes
29	Travelling	Job Searching	Application Engineer	APPLICATION ENGINEER - Manufacturer Client, 16th Feb 2013 Un	18:49:34	2013:3:3	Yes

Table 9.1.2 Travelling Database

## Registration Database Structure:

Table Design: signup @history (localhost)

Fields | Indexes | Foreign Keys | Triggers | Options | Comment

Name	Type	Length	Decimals	Allow Null	
uid	int	100	0	<input type="checkbox"/>	
name	varchar	100	0	<input type="checkbox"/>	
user	varchar	100	0	<input type="checkbox"/>	
pass	varchar	100	0	<input type="checkbox"/>	
email	varchar	100	0	<input type="checkbox"/>	
mobile	varchar	100	0	<input type="checkbox"/>	
date	varchar	100	0	<input type="checkbox"/>	

Fig. 9.1.3 Registration Database Structure

## Registration Database:

Table: signup @history (localhost)

File | Edit | View | Window

Memo | Hex | Image | Sort Ascending | Sort Descending | Remove Sort

uid	name	user	pass	email	mobile	date
1	Manoj Kumar	manoj	manoj	knkmanoj1991@gmail.com	9866177925	25/02/2013
2	Vani	vani	vani	mvani123@gmail.com	9440182728	25/02/2013
3	Mohan Babu	mohanbabu	mohanbabu	mohanbabu@gmail.com	9000810005	26/02/2013
4	Pavan Kumar	pavankumar	pavankumar	pavankumar@gmail.com	8106349503	26/02/2013

Fig. 9.1.4 Registration Database Structure

## User History Database Structure:

[Table Design] history @history (localhost)

File Edit Window

New Save Save As Add Field Insert Field Delete Field Primary Key Move Up Move Down

Name	Type	Length	Decimals	Allow Null	
id	int	50	0	<input type="checkbox"/>	
user	varchar	100	0	<input type="checkbox"/>	
cat	varchar	225	0	<input type="checkbox"/>	
scat	varchar	225	0	<input type="checkbox"/>	
tit	varchar	225	0	<input type="checkbox"/>	
currenttime	varchar	40	0	<input type="checkbox"/>	
currentdate	varchar	40	0	<input type="checkbox"/>	
status	varchar	30	0	<input type="checkbox"/>	

Fig. 9.1.5 User History Database Structure

## User History Database:

[Table] history @history (localhost)

File Edit View Window

Memo Hex Image Sort Ascending Sort Descending Remove Sort

id	user	cat	scat	tit	currenttime	currentdate	status
39	manoj	Travelling	Pilgrims	Tirupati	20:11:22	2013:3:4	Yes
40	manoj	Travelling	Hotels	bhimas	20:12:14	2013:3:4	Yes
41	manoj	Travelling	Sightseeing	goa	8:45:2	2013:3:5	Yes
43	manoj	Travelling	Colleges	jntu	20:25:42	2013:3:6	Yes
44	manoj	Travelling	Job Searching	online jobs	9:37:37	2013:3:7	Yes

Fig. 9.1.6 User History Database

## **10 REFERENCES**

### REFERENCES:

- [1] "Controlling Multimedia Applications Using Hand Gesture Recognition" by Neha S. Rokade, Harsha R. Jadhav, Sabiha A. Pathan , Uma Annamalai.
- [2] "OpenCV-Python Tutorials Documentation" by Alexander Mordvintsev & Abid K.
- [3] "Enhanced Look Based Media Player With Hand Gesture Recognition" by Harshada Naroliya, Tanvi Desai, Shreeya Acharya, Varsha Sakpal.
- [4] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001
- [5] A Vision based Hand Gesture Interface for Controlling VLC Media Player,Siddharth Swarup Rautaray,Indian Institute of Information Technology,Allahabad,Anupam Agrawal,Indian Institute of Information Technology,Allahabad
- [6] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.
- [7] Mr. Sanjay Mena, "A Study on Hand Gesture Recognition", White Paper.
- [8] Ruslana Makovetsky and Gayane Persian,"Face Detection and Tracking with Web Camera", White paper.