**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

Gnana Sangama,Belgaum-560002,Karnataka



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

ACADEMIC YEAR 2019-2020

III B.Tech - 5th Semester

Database System

**PROJECT REPORT**

On

**~!-CLONE~**

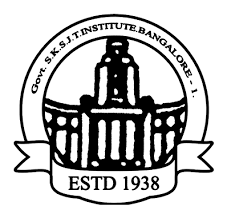
[Photo Sharing and Chatting Application]

Developed by:

|  |  |
| --- | --- |
| **Student Name:** | **Usn:** |
| Lekhana D | 1SK17CS014 |
| Vishwas M Gowda | 1SK17CS039 |

Under the Guidance of :

Mr. Naveen Kumar T.H , Dept of CSE , GSKSJTI



**Govt.Sri Krishnarajendra Silver Jubilee Technological Institute**

(Affiliated to Visvesvaraya Technological University ,Belgaum)

K.R Circle, Bangalore-560001

**ACKNOWLEDGEMENT**

A unique oppurtunity like this come very rarely.It is indeed a pleasure for me to have worked on this project .The satisfaction that accompanises the successful completion of this project is incomplete without the mention of the people whose guidance and suppport are made it possible for me to complete this project.

I express my sincere gratitude to **Dr. K.G. CHANDRASHAKER**, the principal Govt. SKSJTI for his valuable guidance, suggestion and consistent encouragement during the course of my mini project work and timely assistance for completion of the project.

I thank to entire faculty of the Computer Science and Engineering Department, especially **Dr.T.G BASAVARAJU**, Head of the Department CSE, who has given me confidence to believe in myself and complete the project.

I’m grateful to **Prof. NAVEEN T.H,** my internal guide, Computer Science and Engineering Department, who has been a reservoir of immense support and encouragement throughout the process.

Last but not least I would also like to thank my parents and friends for their moral support.

**ABSTRACT**

**!-Clone,** the photo sharing social network service which will helps share Photos with friends.Some people are very much excited in sharing their opinion ,Photos,presenting their skills,work ,this will be open platform for those people. We can write whatever we want and post our random clicks rather than waiting for anyone to do that.

Also most of the people are busy in their schedule, This website will help them to get connected with their friends.They can view their friends Photos , like and Comment on their photos.

People can Upload their Photos with Location and Caption describing their status.People have their First Name,Last Name, User Name ,Date of Birth , e-mail ,Profile-picture and also Uploaded photos in their particular Account.This will be one of the most interesting projects that one can work on and implement.

**Table of Contents**

[CHAPTER 1: **INTRODUCTION**](#_gjdgxs)

[**1.1**](#_30j0zll) **​INTRODUCTION TO DATABASE MANAGEMENT SYSTEM​**

[**1.2**](#_1fob9te) **APPLICATIONS OF DBMS**

[**1.3**](#_1y810tw) **I-CLONE**

[**1.4**](#_4i7ojhp) **AIM OF THIS PROJECT**

[**1.5**](#_2xcytpi) **OBJECTIVES**

[CHAPTER 2](#_tyjcwt)

[2](#_3dy6vkm) **. FUNCTION DESCRIPTION**

[**2.1**](#_1t3h5sf) **HTML​**

[**2.2**](#_4d34og8) **MySQL​**

[**2.3**](#_2s8eyo1) **NodeJS- ExpressJS**

[**2.**](#_2s8eyo1)**4**E-**JS**

[**2.**](#_2s8eyo1)**5****CSS**

[**2.**](#_2s8eyo1)**6****JavaScript**

[**2.**](#_2s8eyo1)**7****BootStrap**

[CHAPTER](#_26in1rg) 3

3 **HARDWARE AND SOFTWARE REQUIREMENTS ​**

[**3.1**](#_1ci93xb) **SOFTWARE REQUIREMENTS: -​**

[**3.2**](#_3whwml4) **HARDWARE REQUIREMENTS: -​**

[CHAPTER 4](#_26in1rg)

[4](#_lnxbz9) **DESIGN AND IMPLEMENTATION​**

**4.1 ​SYSTEM DESIGN:**

[**4.2**](#_1ksv4uv) **​DATABASE DESIGN:**

[**4.3**](#_44sinio) **​TABLE CREATION:**

[**4.4**](#_z337ya) **ER DIAGRAM:**

[**4.5**](#_3j2qqm3) **DATA FLOW DIAGRAM:**

[**4.6**](#_2bn6wsx) **STORED PROCEDURE:**

[4.7](#_2bn6wsx) **TRIGGER:**

[CHAPTER 5](#_2p2csry)

[**5**](#_147n2zr) **CONCLUSION:**

[CHAPTER 6](#_3o7alnk)

[6](#_3o7alnk) **REFERENCES:**

# CHAPTER 1

## **1.INTRODUCTION**

## **1.1 INTRODUCTION TO DATABASE MANAGEMENT SYSTEM**

Databases and database technology have a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic, commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what a database is. Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database. The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following implicit properties: A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse (UoD). Changes to the mini world are reflected in the database. A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database. A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested. A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. The database definition or descriptive information is also stored by the DBMS in the form of a database catalogue or dictionary; it is called meta-data. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

**1.2 APPLICATIONS OF DBMS**

Applications where we use Database Management Systems are: Telecom: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond. Industry: Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example, distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture. Banking System: For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems. Education sector: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees detail etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner. Online shopping: You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

## **1.3. INTRODUCTION TO I-CLONE**

!-clone is an photo sharing and Chatting Social Networking service.

The Profile tab presents all of the user’s photo post in one palce and a short bio. To post a photo ,

click on Camera tab , which will give the option choosing a photo from the device’s library.

Online Photo Sharing , Social Networking service allows the user to share photos with their friends and people with common interest.

It has different tabs to navigate within the app: Profile tab , Camera tab , Home tab, message tab , Search tab.

Profile tab shows the user Bio and instagram Photos. It also where user can edit their bio and account settings.

Camere tab allows the user to share photos from their device image library.

The home tab shows a feed of photos posted by the user and his friends. The user can like , comment on the photos in the feed.

Users can upload photographs and can see other users feed and geotag images with the name of a location.

Feed tab shows the uploaded images of the user and his friends . We can like , comment and can view the photos.

# CHAPTER 2

# 2. FUNCTION DESCRIPTION

* 1. **HTML**

What is HTML?

* HTML is the standard markup language for creating Web pages.
* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup
* HTML elements are the building blocks of HTML pages
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
* Browsers do not display the HTML tags, but use them to render the content of the page
* HTML Documents
* Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.
  1. **MySQL**

What is a Database?

* A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.
* Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.
* Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys.
* A Relational Database Management System (RDBMS) is a software that –
* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL doesn't support SQL check constraints.

**2.3 E-JS**

What is EJS?

* EJS simply stands for Embedded Javascript. It is a simple templating language/engine that lets its user generate HTML with plain javascript. It offers an easier way to interpolate (concatenate) strings effectively.
* EJS is mostly useful whenever you have to output html with a lot of javascript, if you’re dealing with generating dynamic contents or offering something that has to do with real time updates.
* EJS or Embedded Javascript Templating is a templating engine used by Node.js.
* Template engine helps to create an HTML template with minimal code. Also, it can inject data into HTML template at the client side and produce the final HTML.

**2.4 Bootstrap**

* Bootstrap is a free front-end framework for faster and easier web development.Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins.Bootstrap also gives you the ability to easily create responsive designs
* It contains mobile first styles throughout the entire library, instead of using them in the separate files.With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
* It is supported by all popular browsers and its responsive CSS adjusts to Desktops, Tablets and Mobiles.Provides a clean and uniform solution for building an interface for developers.

**2.5 CSS**

* CSS stands for Cascading Style Sheets.
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media.CSS saves a lot of work. It can control the layout of multiple web pages all at once.External stylesheets are stored in CSS files
* Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page.
* Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

**2.6 Node JS**

* Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine).
* Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications.
* Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.
* Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications.
* Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.Node.js = Runtime Environment + JavaScript Library.
* All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. Express provides a minimal interface to build our applications.
* It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express.

**2.7 JavaScript**

* Javascript is a lightweight , High-level ,interpreted programming language.It is designed for creating network-centric applications. It is complimentary to and integrated with java.
* JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.Js has curly-bracket syntax, dynamic -typing , prototype-based, object-oriented and first-class functions. Alongside HTML and CSS , JavaScript is one of the core technologies of the World Wide Web.

**Hardware and Software Requirements:**

**Software Requirements: -**

1. SYSTEM : MacBook Pro.
2. OPERATING SYSTEM : macOS.

DATABASE : MYSQL

WEB BROWSER : Chrome

## 

## **Hardware Requirements: -**

1. PROCESSOR : Intel Core i7.
2. HARD DISK : 1TB.
3. RAM : 8GB.
4. MONITOR : AMD Radeon HD 6750M.
5. MOUSE : TRACK PAD.
6. GRAPHICS : AMD Radeon HD 6750M 512 MB
7. Intel HD Graphics 3000 512 MB.

**Front and Back end:-**

1. FRONT-END : E-JS, HTML, CSS, BOOTSTRAP 4, JAVA SCRIPT.
2. BACK-END : NODE-JS, MY-SQL.

CHAPTER 4

# DESIGN AND IMPLEMENTATION

# System Design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel.

**System design goes through two phases of development:**

* Logical Design
* Physical Design.

## **Logical Design**

The logical flow of a system and define the boundaries of a system. It includes the following steps:

* Reviews the current physical system – its data flows, file content, volumes, frequencies etc.
* Prepares output specifications – that is, determines the format, content and frequency of reports.
* Prepares input specifications – format, content and most of the input functions.
* Prepares edit, security and control specifications.
* Specifies the implementation plan.
* Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
* Reviews benefits, costs, target dates and system constraints.

## **Physical Design**

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

* Design the physical system.
* Specify input and output media.
* Design the database and specify backup procedures.
* Design physical information flow through the system and a physical design Walk through.
* Plan system implementation.
* Prepare a conversion schedule and target date.
* Determine training procedures, courses and timetable.
* Devise a test and implementation plan and specify any new hardware/software.
* Update benefits, costs, and conversion date and system constraints.

**Database design**

The database that is used to design the web application is MySQL. MySQL workbench is used to create tables and run queries. In this application development, we have used MySQL to store user details, event details, location details, food details, decoration details.

Hence, we have identified five tables to achieve desired functionality.

i. User\_table: holds details of Admin.

ii. Login : holds details of User.

iii. Photos : holds details of events booked by the user.

iv. Likes: holds details of Likes got to the post.

v. comment table: holds details of comments posted.

vi. message table : holds the Messages sent by The respected User.

**TABLE CREATION:**

**1. User Names Table**

CREATE TABLE `User\_Name` (

`U\_Id` int(11) DEFAULT NULL,

`User\_Name` varchar(20) NOT NULL,

`F\_Name` varchar(20) DEFAULT NULL,

PRIMARY KEY (`User\_Name`),

KEY `U\_Id` (`U\_Id`),

CONSTRAINT `user\_name\_ibfk\_1` FOREIGN KEY (`U\_Id`) REFERENCES `user\_details` (`U\_Id`)

)

|  |  |
| --- | --- |
| Name | Datatype |
| U\_id | int(11) |
| User\_Name | varchar(20) |
| F\_name | varchar(20) |

**2. User Details Table**

CREATE TABLE `user\_details` (

`U\_Id` int(11) NOT NULL,

`U\_Name` varchar(20) DEFAULT NULL,

`F\_Name` varchar(20) DEFAULT NULL,

`L\_Name` varchar(20) DEFAULT NULL,

`Email\_Id` varchar(30) DEFAULT NULL,

`Gender` char(1) DEFAULT NULL,

`P\_pic` blob,

`D\_O\_B` date DEFAULT NULL,

`D\_O\_Creation` date DEFAULT NULL,

`Bio` text,

PRIMARY KEY (`U\_Id`),

UNIQUE KEY `U\_Id` (`U\_Id`)

)

|  |  |
| --- | --- |
| Name | Datatype |
| U\_Id | Int(11) |
| U\_Name | Varchar(20) |
| F\_Name | Varchar(20) |
| L\_Name | Varchar(20) |
| Email\_Id | Varchar(20) |
| Gender | Char(1) |
| P\_Pic | Blob |
| D\_O\_B | Date |
| D\_O\_Creation | Date |
| Bio | text |

**3. Photo Creation Date Table**

CREATE TABLE `Photo\_Cre\_date` (

`photo\_id` int(11) DEFAULT NULL,

`Date\_of\_Cre` datetime DEFAULT CURRENT\_TIMESTAMP

)

|  |  |
| --- | --- |
| Name | Datatype |
| Photo\_id | Int(11) |
| Date\_of\_Cre | datetime |

**4. Friends Table**

CREATE TABLE `friends` (

`U\_Id` int(11) DEFAULT NULL,

`Friend\_Id` int(11) NOT NULL,

`Friends\_Name` varchar(20) DEFAULT NULL,

PRIMARY KEY (`Friend\_Id`),

KEY `U\_Id` (`U\_Id`),

CONSTRAINT `friends\_ibfk\_1` FOREIGN KEY (`U\_Id`) REFERENCES `User\_Details` (`U\_Id`)

)

|  |  |
| --- | --- |
| Name | Datatype |
| U\_Id | Int(11) |
| Friend\_Id | Int(11) |
| Friends\_Name | Varchar(20) |

**5. Photo Details Table**

CREATE TABLE `photos` (

`photo\_id` int(11) NOT NULL AUTO\_INCREMENT,

`U\_Id` int(11) DEFAULT NULL,

`Caption` text,

`Location` varchar(30) DEFAULT NULL,

`Photo\_Name` varchar(20) DEFAULT NULL,

PRIMARY KEY (`photo\_id`),

KEY `U\_Id` (`U\_Id`),

CONSTRAINT `photos\_ibfk\_1` FOREIGN KEY (`U\_Id`) REFERENCES `User\_Details` (`U\_Id`)

)

|  |  |
| --- | --- |
| Name | Datatype |
| Photo\_id | Int(11) |
| U\_Id | Int(11) |
| Caption | Text |
| Location | Varchar(20) |
| Photo\_Name | Varchar(20) |

**6. Comment Table**

CREATE TABLE `comment` (

`U\_Id` int(11) DEFAULT NULL,

`photo\_id` int(11) DEFAULT NULL,

`comment` text,

`Time` time DEFAULT NULL,

`C\_Name` varchar(20) DEFAULT NULL

)

|  |  |
| --- | --- |
| Name | Datatype |
| U\_Id | Int(11) |
| Photo\_id | Int(11) |
| Comment | Text |
| Time | Time |
| C\_Name | Varchar(20) |

**7. Likes Table**

CREATE TABLE `likes` (

`U\_Id` int(11) DEFAULT NULL,

`photo\_id` int(11) NOT NULL AUTO\_INCREMENT,

`Likes` int(11) DEFAULT NULL,

KEY `U\_Id` (`U\_Id`),

KEY `photo\_id` (`photo\_id`),

CONSTRAINT `likes\_ibfk\_1` FOREIGN KEY (`U\_Id`) REFERENCES `User\_Details` (`U\_Id`),

CONSTRAINT `likes\_ibfk\_2` FOREIGN KEY (`photo\_id`) REFERENCES `photos` (`photo\_id`)

)

|  |  |
| --- | --- |
| Name | Datatype |
| U\_Id | Int(11) |
| Photo\_id | Int(11) |
| Likes | Int(11) |

**8. Messages Table**

CREATE TABLE `Messages` (

`U\_Id` int(11) NOT NULL,

`U\_Name` varchar(20) DEFAULT NULL,

`Messages` text,

`M\_Id` int(11) NOT NULL,

KEY `U\_Id` (`U\_Id`),

CONSTRAINT `messages\_ibfk\_1` FOREIGN KEY (`U\_Id`) REFERENCES `User\_Details` (`U\_Id`)

)

|  |  |
| --- | --- |
| Name | Datatype |
| U\_Id | Int(11) |
| U\_Name | Varchar(20) |
| Messages | text |
| M\_Id | Int(11) |

**DATAFLOW DIAGRAM:**

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process-oriented system flowchart. For these reasons DFD’s are often referred to as logical data flow diagrams.

**ER DIAGRAM:**

**STORED PROCEDURE:**

A Stored Procedure is a set of SQL statements with an assigned name, which are stored in a relational database management system such that it can be reused and shared by multiple programs.

Name : add\_date\_entry

Type: Procedure

**DELIMITER $$**

**CREATE PROCEDURE add\_date\_entry(IN id INT(11))**

**BEGIN**

**INSERT INTO `photo\_cre\_date`(`photo\_id`) VALUE(id);**

**END$$**

**TRIGGER:**

A Trigger is a special kind of stored procedure that executes automatically when an event occurs in Database Server.

Trigger : main\_trigger

Time-Event : AFTER INSERT

**CREATE TRIGGER `main\_trigger`**

**AFTER INSERT**

**ON `iclone`.`photos`**

**FOR EACH ROW**

**BEGIN**

**CALL add\_date\_entry(NEW.photo\_id);**

**END$$**

**DELIMITER ;**

**FRONT-END CODE:**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<link type="text/css" rel="stylesheet" href="css/materialize.min.css" media="screen,projection"/>

<meta name="viewport" initial-scale=1.0"/>

<title>

News Feed

</title>

<link rel="stylesheet" type="text/css" href="Feed.css">

</head>

<body align="center" class="allelements">

<div class = "wrapper">

<div class = "flex-wrapper">

<div class = "flex-wrapper-items">

<a href="http://localhost:8080/upl1">

<span style="color: black" class="glyphicon glyphicon-camera"></span>

</a>

</div>

<div class = "flex-wrapper-items">!-Clone</div>

<div class = "flex-wrapper-items">

<a href="http://localhost:8080/message1">

<span style="color: black" class="glyphicon glyphicon-comment"></span>

</a>

</div>

</form>

</div>

</div>

<div class = "sub-wrapper">

<div class = "stories-flex">

<div class = "stories-flex-text">Friends</div>

</div>

</div>

<div class = "friend-stories">

<div class = "friend-stories-wrapp">

<% path1.forEach(function(abc){%>

<div class = "photo-friend-text-name">

<img src = "<%= abc.P\_Pic %>" class = "photo-friend" width = "80px" alt = "error"/>

<a style="color: black;" href="http://localhost:8080/<%=abc.U\_Name %>">

<div class = "this-photo-friend-text-name"><%=abc.U\_Name %></div>

</a>

</div>

<%})%>

</div>

</div>

</br>

-------------------------------------------------------------------------------------------------------------------------------------------------

<%path.forEach(function(img){%>

<div class = "main">

<div class = "for-main">

<div class = "for-main-two">

<img src = "<%= img.P\_Pic%>" class = "logo-small" width = "50px" alt = "error"/>

<div class = "for-main-text s"><%= img.U\_Name %></div>

<div class = "for-main-text s">---- <%= img.location %></div>

<div class = "for-main-text s">

</div>

</div>

</div>

<div class = "after-main">

<img src= '<%= img.photo\_Name%>' height="100%" width="100%"></br>

<div class = "likes">

<div class = "sub-likes">

<i class="k fa fa-heart"></i>

<i class="l fa fa-comment"></i>

<a href="http://localhost:8080/<%=img.photo\_name%>">

<span style>-----------------------------------------------------------------

</div>

</div>

<div class = "watch" align="left">Likes:<%= img.likes %> </div>

<div class = "sub-main" align="left"> <span class = "black-sub-main"><%= img.U\_Name %> : </span><%= img.caption %> </div>

<div class = "timing" align="left"> <span class = "time-this"></span> <%= img.date\_Of\_Cre %></div>

<div class = "sub-main" align="left"> <span class = "black-sub-main"><%= img.C\_Name %> : </span><%= img.comment %> </div>

</div>

</div>

------------------------------------------------------------------------------------------------------------------------

<%})%>

<div class= "bott">

<div class = "panel-footer">

<div class = "tabs footer-items">

<a href="http://localhost:8080/feed1">

<span style="color: black" class="glyphicon glyphicon-home"></span>

</a>

</div>

<div class = "flex-wrapper-items">

<a href="http://localhost:8080/search1">

<span style="color: black" class="glyphicon glyphicon-search"></span>

</a>

</div>

<div class = "flex-wrapper-items">

<a href="http://localhost:8080/account1">

<span style="color: black" class="glyphicon glyphicon-user"></span>

</a>

</div>

</div>

</body>

<script>

window.onload = function() {

var i;

var like = document.getElementsByClassName("k");

var foot = document.getElementsByClassName("footer-items");

for (i = 0; i < like.length; i++) {

like[i].addEventListener("click", function() {

this.classList.toggle("red");

});

}};

</script>

</html>

**DATABASE CONNECTIVITY:**

var express = require('express');

var ejs = require('ejs');

var app = express();

var path = require('path');

var mysql = require("mysql");

app.set('view engine','ejs');

app.use(express.static('public'));

var con = mysql.createConnection({

host : "localhost",

user : "root",

password : "vishu@1234",

database : "iclone"

});

con.connect(function(err){

if (err)

throw err;

console.log("Connected to MySql database");

console.log("Started accessing the requirements");

console.log("Fetched All the Results");

console.log("Try http://localhost:8080/ for Login")

});

app.get('/feed1',(req,res)=>{

var sql = 'select User\_Details.U\_Name,User\_details.P\_Pic,User\_details.Bio,photos.photo\_id, photos.caption, photos.location,photos.photo\_Name,photo\_Cre\_Date.date\_Of\_Cre,likes.likes,comment.comment,comment.

Time,comment.C\_Name from User\_Details,photos,photo\_Cre\_date,likes,comment where user\_details.U\_Id = photos.U\_Id and photos.photo\_id = likes.photo\_id and photos.photo\_id= Comment.photo\_id and photos.photo\_id = Photo\_cre\_Date.photo\_id order by photos.photo\_id desc;

con.query(sql,(err,results,fields)=>{

if (err)

throw err;

var sqly = 'select U\_Name,P\_Pic from User\_Details;'

con.query(sqly,(err,result,fields)=>{

if (err)

throw err;

console.log(result);

res.render('Feed1',{

path:results,

path1:result

});

});

});

});

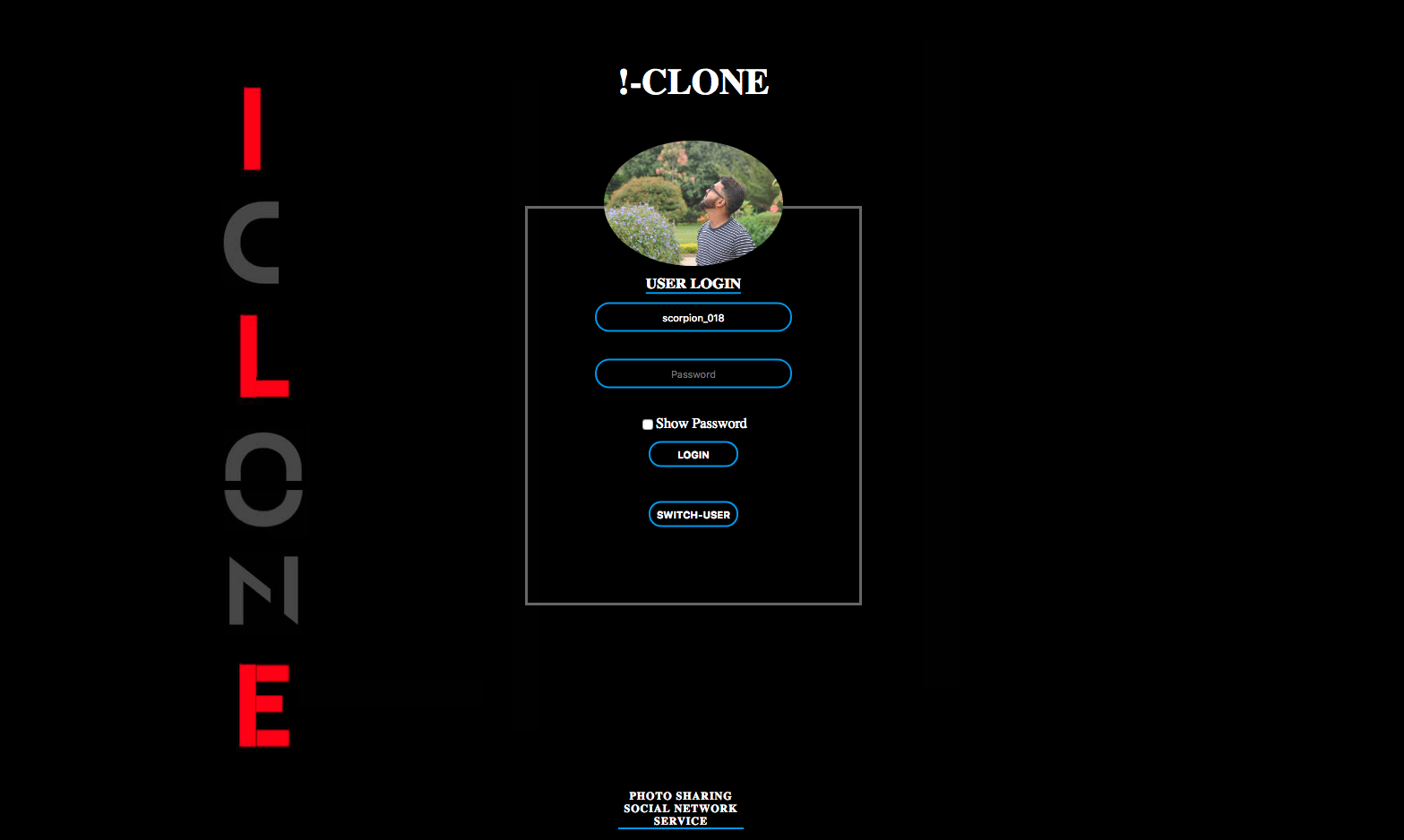
app.listen(port,()=>{

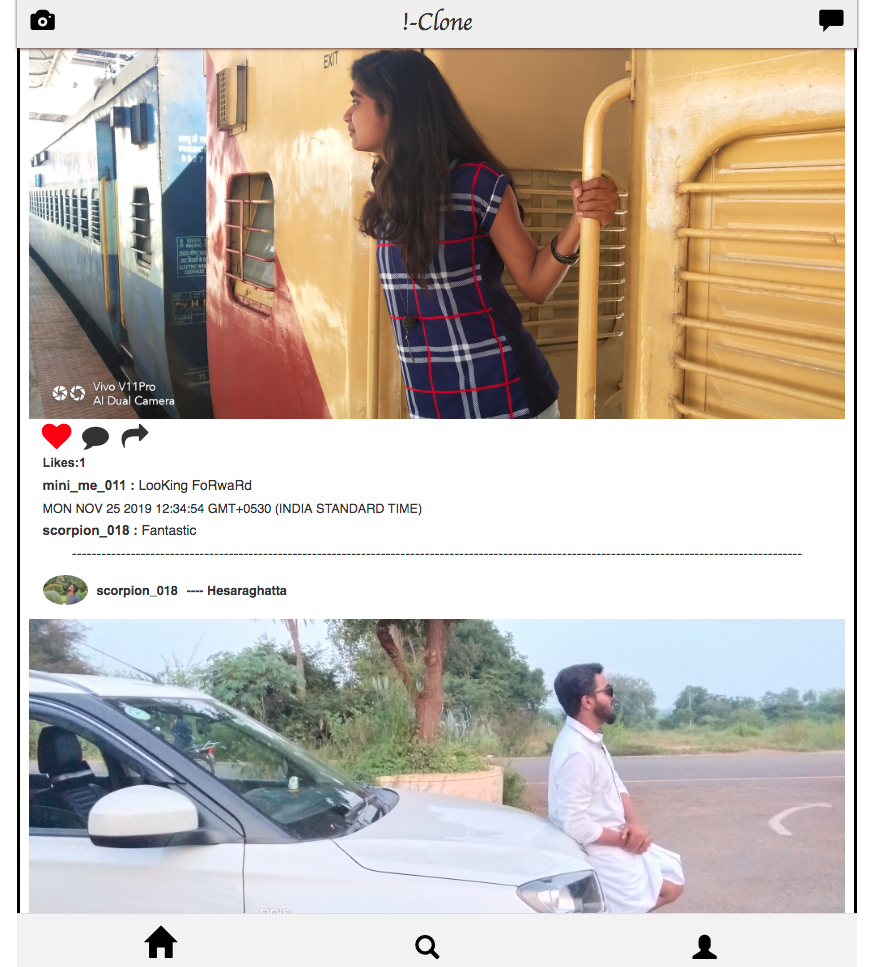
console.log("Listeining to port 8080");

});

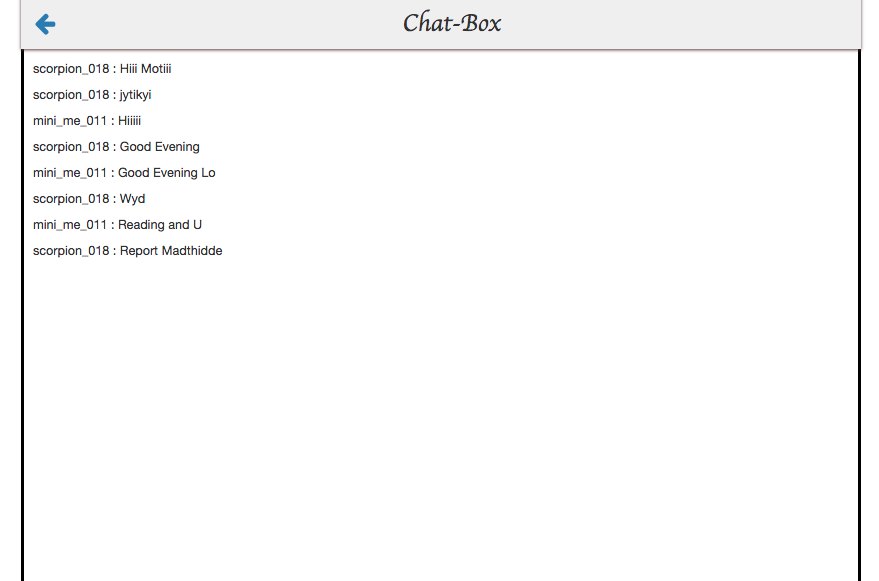
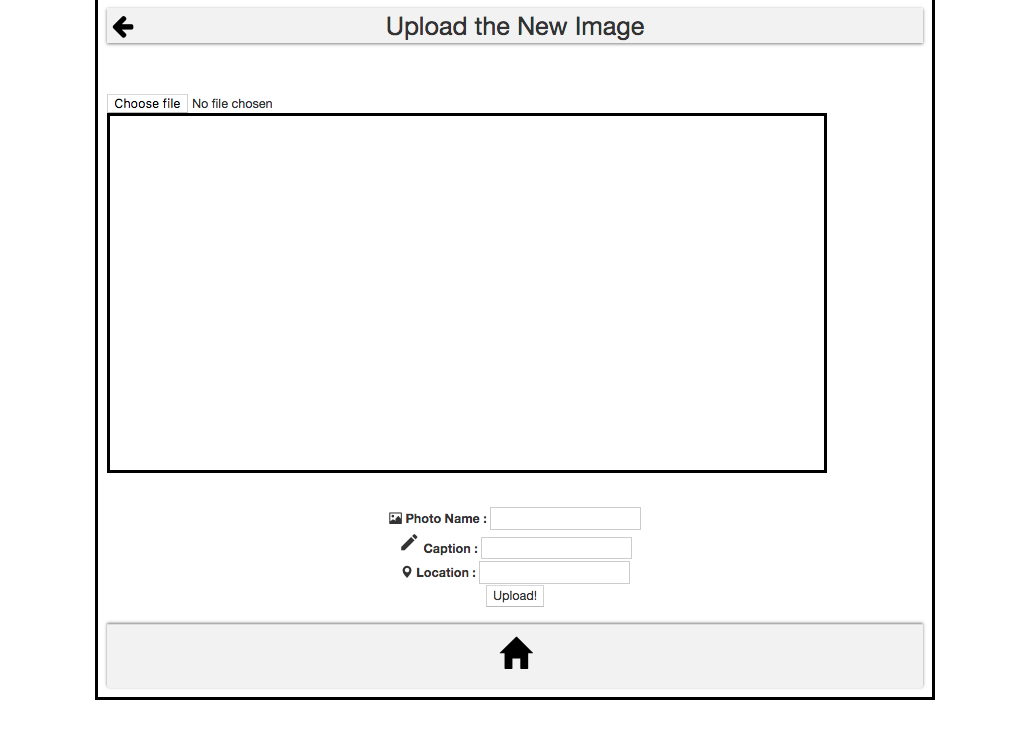
**USER INTERFACE**

**Login Page:**

****

**Feed Page:**

**Upload Page and Chat Box:**

**Account Page:**

**CONCLUSION:**

**REFERENCES:**