**1. INTRODUCTION**

**1.1 ABOUT THE PROJECT**

In this project, with the increase of missing complaints in today's society, the task of police to find these complaints becomes very difficult. This is despite the fact that missing cases have arrived at the station. Tracking a call is difficult and takes a lot of effort and time. While cc footage can be used in some cases, attempts are required from each video recording included in the file. This is time consuming and difficult work. Every day in our society, there are children, youth, young women, the mentally handicapped, the elderly with dementia, etc. Countless people go missing. Although the police department filed a lawsuit against them. Often times they are very difficult to find. According to the current system, if a person is found missing, we must report his whereabouts to the nearest police station. After the complaint, the police will start an investigation by obtaining the necessary information.

This is a time-consuming process that requires a lot of effort. In the world, a countless number of people are missing every day which includes kids, teens, mentally challenged, old-aged people with Alzheimer's, etc. Most of them remain untraced. In this project, file verification process in other words document verification is very important nowadays for a lot of places. It actually allows validating the info provided by their public are true and correcting without any discrepancies.  The public will register and have a dashboard which will show public personal details in this web application. Once registered public will have access the different files are uploading and download options. One person is unfortunately missing in the anywhere and any place. User is getting the person in anywhere and searches the missing person’s details (such as ID number) in this website. Once search the ID number it will display the person’s personal details.

**1.2 MODULE DESCRIPTION**

**Admin login**

In this module maintain the admin login details. Admin is used unique username and password. They are only access in this module such as add, update and delete. Other users doesn’t access in this project.

**View public details**

Admin can view the public details. The module maintain to the public add certificate.

**View certificate details**

Admin can view the how many publics are register to our web application and helps to the missing person. They module maintain to the public certificates details.

**Public registration**

In this module maintain the public registration details. It contains public id, public name, mobile no, password, mail id and address. Publics are stored in our personal details.

**Public login**

Public are get unique user name and password in this module after registration. Login details are stored in this module. Other users doesn’t access in this module. It includes username and password.

**Add ID proof details**

In this module public can add their details and ID proofs. The module maintain by the public users. Registered users will have access the different files are upload in this section. Download files are manage to the different users are download to our document in any time and also admin can get missing files or persons and searches that missing details are view in this web application. It includes public register number, ID name, ID number etc.

**User registration**

In this module used to maintain the user registration details. It contains user id, user name, mobile no, password, mail id and address.

**Search ID number**

In this module, if any person or miss their certificate in anywhere and searches the missing public details (such as ID number) in this website. Once search the ID number it will display the person’s personal details.

**1.3 SYSTEM SPECIFICATION**

**1.3.1 HARDWARE SPECIFICATION**

System : Intel core2dual

Hard Disk : 500 GB

Floppy Drive : 1.44 MB

Monitor : 14’ Colour Monitor

Mouse : Optical Mouse

Ram : 4 GB

**1.3.2 SOFTWARE SPECIFICATION**

Operating system : Windows 10

Front End : PHP

Back End : MYSQL

**1.4 SOFTWARE FEATURES**

**About PHP**

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP widely used; free and efficient alternative to competitors such as Microsoft’s ASP.PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is similar to pearl and C.

PHP is open source that it is readily available and absolutely free. Stability, flexibility and speed are chief qualities that attract to choose PHP.PHP have multiple extensions and is extremely scalable.

**Server-side scripting**

This server-side scripting is the most traditional and main target field for PHP. Programmer needs three things to make this work. Programmer need to run the web server, with a connected PHP installation. Programmer can access the PHP program output with a web browser, viewing the PHO page through the server. All these can run on your home machine if programmers are just experimenting with PHP programming.

**Command line scripting**

Programmer can make a PHP script to run it without any server or browser. Programmers only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using croon (on\*nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.

**Features of PHP**

* PHP runs on different platforms (Windows, Linux, UNIX, etc.)
* PHP is compatible with almost all servers used today.
* PHP is free to download from the official PHP resource: www.php.net.

**About MYSQL**

MYSQL is an open-source relational database management system (RDBMS) is developed, distributed and supported by MYSQL AB. MYSQL is a popular choice of database for use in web applications MYSQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory. MYSQL is easy to use, yet extremely powerful, secure, and scalable. And because of its small size and speed, it is the ideal database solution for Web sites.

**MYSQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporation network. To add, access and process data stored in a computer database we need a database management system such as MYSQL server. Since computers are very good at handling large amount of data, database management system plays a central role in computing.

**MYSQL is a relational database management system**

A relational database stores separate data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MYSQL” stands for “Structured Query Language”. SQL is the most common standardize language used to access database and is defined by the ANSI/ISO SQL standard. The SQL standard has been evolving since 1986 and several versions exist.

**MYSQL software is open source**

Open source means that it is possible for anyone to use modify the software. Anybody can download the MYSQL software uses the GPL (GNU General Public License), to define what we may and may not use do with the software.

**MYSQL Server works in Client/ Server or embedded systems**

The MYSQL database software is a client/server system that consists of a multi-threaded SQL server that supports different backend, several different client programs and libraries, administrative tools and a wide range of Application Programming Interface (APIs). A large amount of contributed MYSQL software is available:

Modern day websites seem to be relying more and more on compel the Structured Query Language is a very popular database language, and its standardization makes it easy to store, update and access data. One of the most powerful SQL servers out there is called MYSQL and surprisingly enough, it’s free.

Some of the features of MYSQL include: Handles large databases, in the area of 50,000,000+records. No memory leaks. Tested with a commercial memory leakage detector (purify). A privilege and password system which is very flexible and secure, and which allows host-based verification. Passwords are secure since all password traffic when connecting the server is encrypted.

**Features of MYSQL**

**Client/server Architecture:** MYSQL is a client/server system. There is a database server (MYSQL) and arbitrarily many clients (application programs), which communicate with the server. The clients can run on the same computer as the server or on another computer.

**SQL Compatibility:** As before said SQL is a standardized language for querying and updating data and for the administration of a database. Through the configuration setting sol-mode we can make the MYSQL server behave for the most part compatibly with various database systems.

**Stored procedures:** Stored procedures (SPs for short) are generally used to simplify steps such as inserting or deleting a data record.

**Triggers:** Triggers are SQL commands that are automatically executed by the server in certain database operations INSERT, UPDATE, and DELETE, MYSQL has supported triggers.

**Replication:** Replication allows the contents of a database to be copied (replicated) onto a number of computers to increase protection against system and to improve the speed of database queries.

**Platform independence:** MYSQL can be executed under a number of operating systems. The most important are Apple Macintosh OS X, Linux, Microsoft Windows, and the Unix.

**Speed:** MYSQL is considered a very fast database program.

**2. SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

This project is developed for maintain the person file verification details. The process of identifying a missing person often involves comparing available information about a person (such as physical characteristics, clothing, and last known location) with records from various sources (such as social media, surveillance footage, and missing persons files). This process requires extensive and complex data analysis, which is difficult and time consuming. To solve this problem now they are looking for better alternative solution.

**2.1.1 DRAWBACKS OF EXISTING SYSTEM**

* Erroneous data entry due to human errors.
* Searching and verifying of exact files are more time consuming.
* Information’s are not in a presentable form.
* It is not user friendly.
* Slow processing.

**2.2 PROPOSED SYSTEM**

According to the current system, if a person is found missing, we must report his whereabouts to the nearest police station. After the complaint, the police will start an investigation by obtaining the necessary information. In this project, file verification process in other words document verification is very important nowadays for a lot of places. It actually allows validating the info provided by their public are true and correcting without any discrepancies.

**2.2.1 ADVANTAGES OF PROPOSED SYSTEM**

* It can store larger amount of data i.e., memory is very large and can have a back up copy when needed.
* Very good at repeated calculations.
* All the process has done at high speed.
* It does the work very efficiently.
* Reduced manual work.

**3. SYSTEM DESIGN AND DEVELOPMENT**

**3.1 FILE DESIGN**

System design is the process of planning a new system to complement or altogether replace the old system. The purpose of the design phase is the first step in moving from the problem domain to the solution domain. The design of the system is the critical aspect that affects the quality of the software. System design is also called top-level design. The design phase translates the logical aspects of the system into physical aspects of the system.

**3.2 INPUT DESIGN**

The data, which is input to a computer – based information system, must be correct. If data is carelessly input and errors enter the system, it will lead to incorrect results whose consequences will be expensive and embarrassing to the designer. In data processing, the data entry operator often makes errors. This can be controlled by input design by using menu, interactive dialogue, consistent format etc.

In this system the users are provided with user friendly pages to give the input and if the user gives any wrong input validations are done and message boxes are provided in the necessary places. The message specified in the message box is specified in a polite and in an informative manner.

System is interactive dialogue, which simplifies the data entry or access, instead of remembering what to enter. User can choose from a list of options and type it in the cursor position. This will reduce the number of corrections while entering the data.

**3.3 DATABASE DESIGN**

The database design involves creation of tables that are represented in physical database as stored files. They have their own existence. Each table constitute of rows and columns where each row can be viewed as record that consists of related information and column can be viewed as field of data of same type. The table is also designed with some position can have a null value.

The database design of project is designed in such a way values are kept without redundancy and with normalized format. Refer the appendix for screen shots of database design.

**3.3.1 TABLE DESIGN**

**Table name: Public registration**

**Primary key: pid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Description** |
| pid | integer | 11 | Public identification |
| name | varchar | 15 | Name |
| img | file | - | Image |
| addr | varchar | 20 | Address |
| mob | integer | 10 | Mobile number |
| mail | varchar | 20 | Mail id |
| pwd | varchar | 15 | Password |

**Table name: Add ID proof details**

**Primary key: perid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Description** |
| perid | integer | 11 | Person identification |
| idno | varchar | 15 | ID number |
| idtype | varchar | 20 | ID type |
| file | file | - | ID file |
| dwnld | file | - | Download file |

**Table name: User registration**

**Primary key: uid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Description** |
| uid | integer | 11 | User identification |
| uname | varchar | 15 | User name |
| addr | varchar | 20 | Address |
| mob | integer | 10 | Mobile number |
| mail | varchar | 20 | Mail id |
| pwd | varchar | 15 | Password |

**3.4 OUTPUT DESIGN**

The proposed system is a web oriented system and hence it does not provide any reports. The output results are viewed in the web pages itself. Outputs from the computer system are required primarily to communicate the result of processing to users. They are also used to override a permanent copy of the results for later consultation.The output reports and input documents should be documented in terms of data content and approximate layout; it is not necessary to define the methods of presentation. It is possible to work back for the output data items are derived by calculations or by logical deduction.

**3.5 SYSTEM DEVELOPMENT**

**TOP DOWN APPROACH**

The importance of new system is that it is user friendly and a better interface with user’s working on it. It can overcome the problems of manual system and the security problem.

Top down approach of software development is the incremental approach to the construction of program structure. Modules are integrated by moving through the control hierarchy, beginning with the main control module. Module subordinate to the main control modules is incorporate into the structure in either a depth first or breadth first manner.

The top down approach is performed in a serious of five steps

1. The main module that is overall software is divided into five modules that are under the control of the main control module.

2. Depending on the top down approach selected subordinate stubs is replaced one at a time with actual components.

3. Tests are conducted as each component is integrated

4. On completion of each test another stub is replaced with real time component.

5. Regression testing may be conducted to ensure the new errors have not been introduced.

**4. SYSTEM TESTING AND IMPLEMENTATION**

**4.1 TESTING**

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

**Types of Testing**

* Unit testing
* Integration testing
* Validation testing
* Output testing
* User acceptance testing
* White box testing
* Black box testing

**4.1.1 Unit Testing**

All modules were tested and individually as soon as they were completed and were checked for their correct functionality.

**4.1.2 Integration Testing**

The entire project was split into small program; each of these single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructors were used. It give a lot of problem by not functioning is an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in frames are convenient and it is satisfied. When the frames where given for the test, the end user gave suggestion. Based on their suggestions the frames where modified and put into practice.

**4.1.3 Validation Testing**

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., Validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

**Output Testing**

After performing the validation testing the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is on screen and another one is printed format. The output format on the screen is found to be corrected as the format was designed in the system phase according to the user needs. And for the hardcopy the output comes according to the specifications requested by the user.

**4.1.4 White box testing**

White box testing(also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/) in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.

Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

Definition by ISTQB

* **White-box testing:** Testing based on an analysis of the internal structure of the component or system.
* **White-box test design technique:** Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.

**4.1.5 Black box testing**

Black box testing, also known as Behavioral Testing, is a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/) in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behavior or performance errors
* Initialization and termination errors

Definition by **ISTQB**

* **Black box testing:** Testing, either functional or non-functional, without reference to the internal structure of the component or system.
* **Black box test design technique:** Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.

**Acceptance testing**

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements. Tools to special importance during acceptance testing include:

**Test coverage Analyzer**

Records the control paths followed for each test case.

**Timing Analyzer**

Also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance.

**4.2 SYSTEM IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system.

According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system. As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

**4.3 SYSTEM MAINTENANCE**

According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system. As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

**5. CONCLUSION AND FUTURE ENHANCEMENT**

**5.1 CONCLUSION**

The main objective of the project is to bring a full-fledged computerized organization, and to enable the transaction details to maintain records, which makes of the employees easier. Thus, the proposed system has been developed with good amount of flexibility without compromising on the response time.

Computerization of the entire system will enhance more accuracy and reduces major part of clerical works. Fast, clear and legible reports can be generated without any ambiguity. Integrated database design and ease of maintenance is a major advantage of the system. User friendliness is a unique feather of the system. Hence by developing a system that is user-friendly in nature, many users are able to work on the system with little of computer knowledge and training.

**5.2 FUTURE ENHANCEMENT**

There is a wide scope for future development of the software. The world of computer fields is not static it is always subject to change. The technology which is famous today will become outdated very next day. To keep abstract of technical improvements, the system may be refinement. So it is not concluded. Yet it will improve with further enhancements.

It is essential to change the software when new software arrives with more advanced feathers. So it is much necessary for further development. Further enhancements can be done in an efficient manner with disruption to the system.

**BIBLIOGRAPHY**

**BOOK REFERENCES**

1. Jesus Castagnetto, Sascha Schumann, “Professional Php Programming”, Addison wisely Publication, Fifth Edition.
2. Jay Greenspan, Brad Bulgar, “Mysql/Php Database Applications”, Tata McGraw-Hill Publishing Company, Third Edition.
3. William Stallings, “Cryptography And Network Security”, Tata McGraw-Hill Publishing Company, Third Edition.
4. Bruce Schneier, “Applied Cryptography”, Pearson Education, Second Edition.
5. Rogers Pressman, “Software Engineering and Applications”, Galgotie Publication, Sixth Edition.

**REFERENCES WEBSITE**

* 1. <www.onlinetutorial.com>
  2. www.cryptography.com
  3. www.tenders.com
  4. www.computerhope.com/starthtm.htm‎
  5. [www.webdesign.about.com/od/webdesignbasics/u/webdesignbasics.htm](http://www.webdesign.about.com/od/webdesignbasics/u/webdesignbasics.htm)
  6. www.w3schools.com/php/php\_mysql\_intro.asp

**APPENDIX**

**A. DATA FLOW DIAGRAM**

**Level 0**

User

Public

Admin

Username Provide

Password

**Level 1**

Username Manage

pub\_db

Admin

Password

id\_db

**Level 2**

Login after registration

reg\_db

Public

Valid

Invalid

User add details

id\_db

**Level 3**

Login after registration

reg\_db

User

Valid

Invalid

User search ID no

id\_db

**B. SYSTEM FLOW DIAGRAM**

Admin

Login

Invalid

Valid

Manage

Public Details

ID proof

Details

User Details

Storage

User Database

Public Database

ID proof

Database

Report

User Report

ID proof Report

Public Report

**C. ENTITY RELATIONSHIP DIAGRAM**

Public registration

Manage

Admin

Process

ID proof details

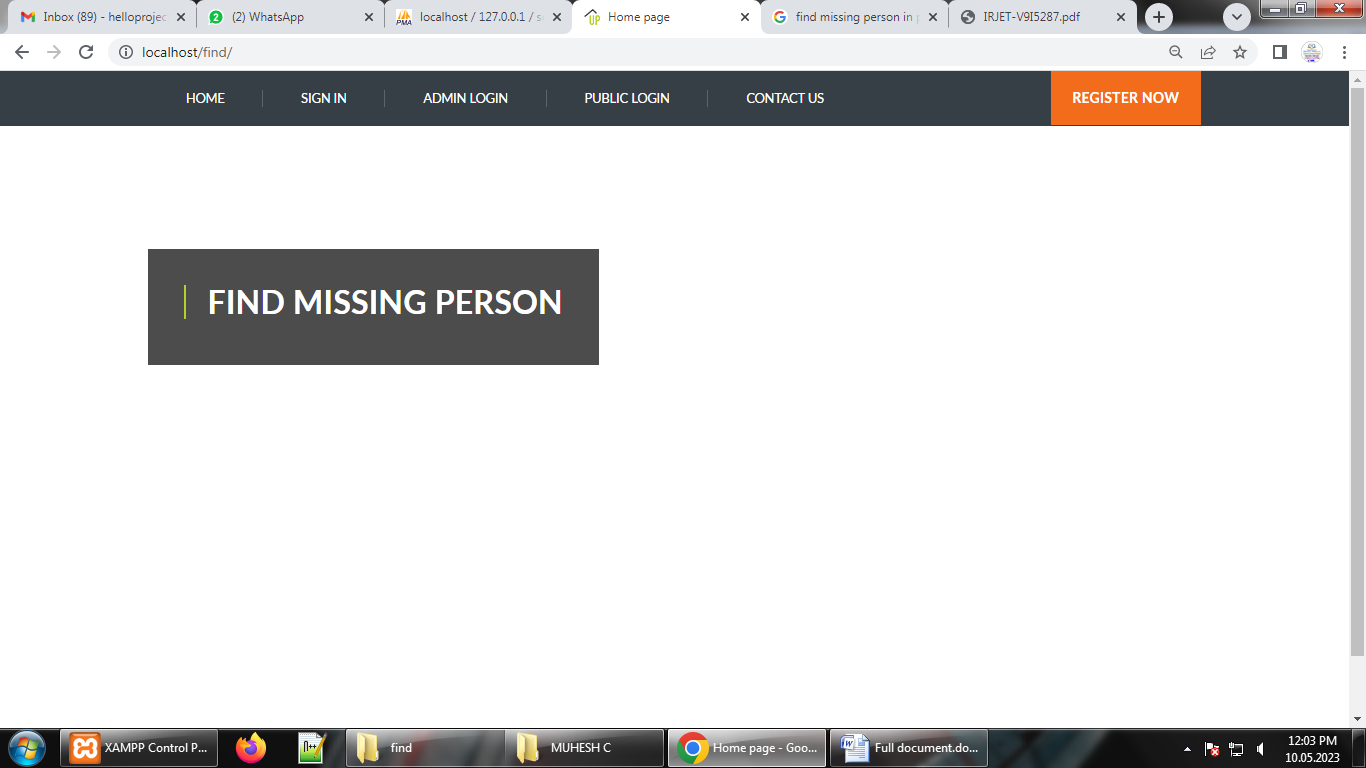
View

Search

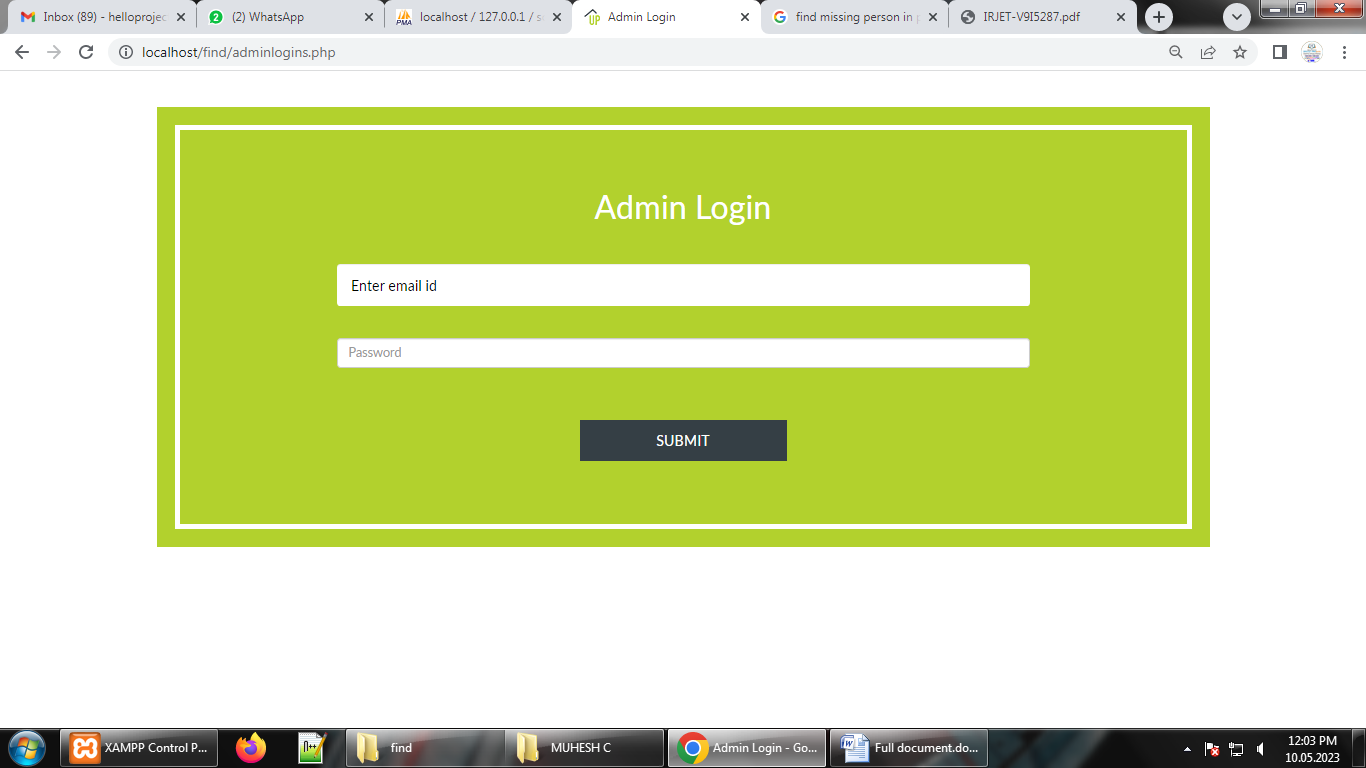
User registration

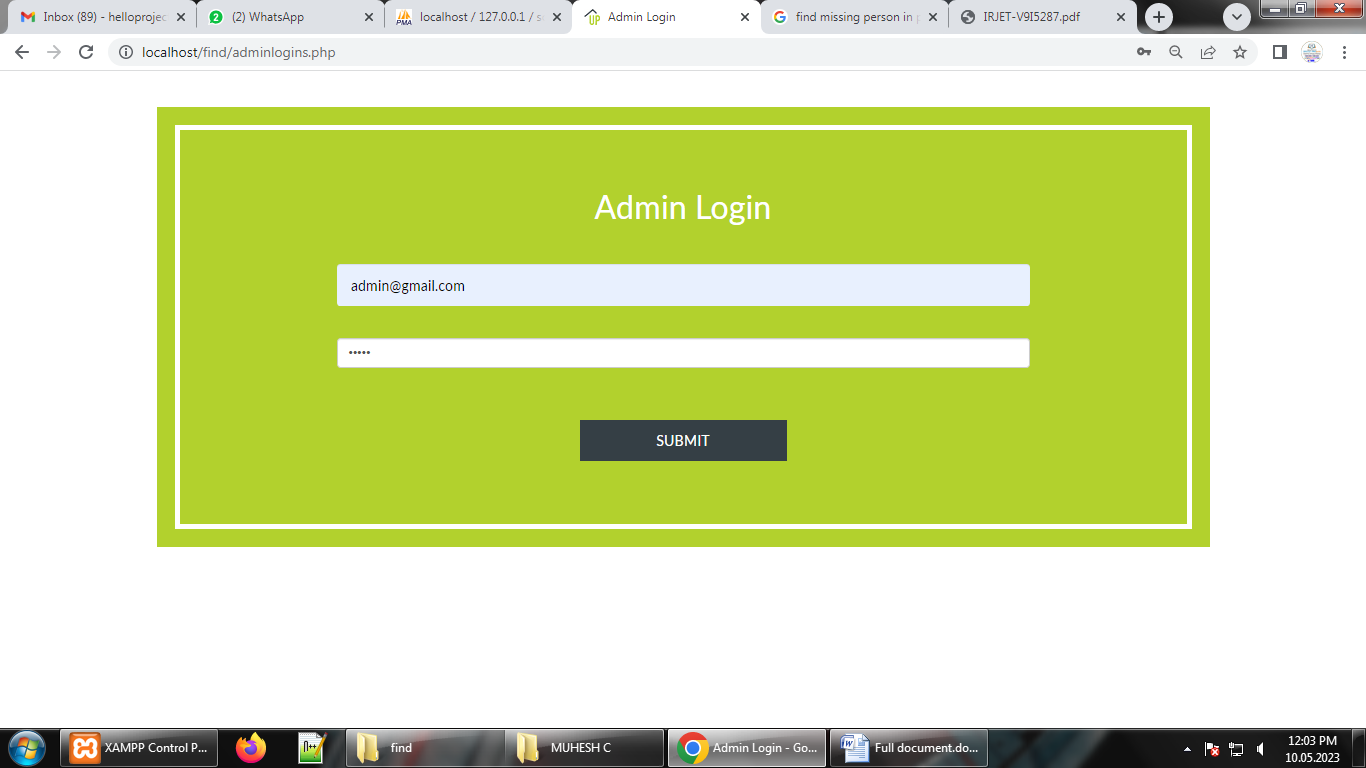
**D. SAMPLE SCREENS**

**Main page**

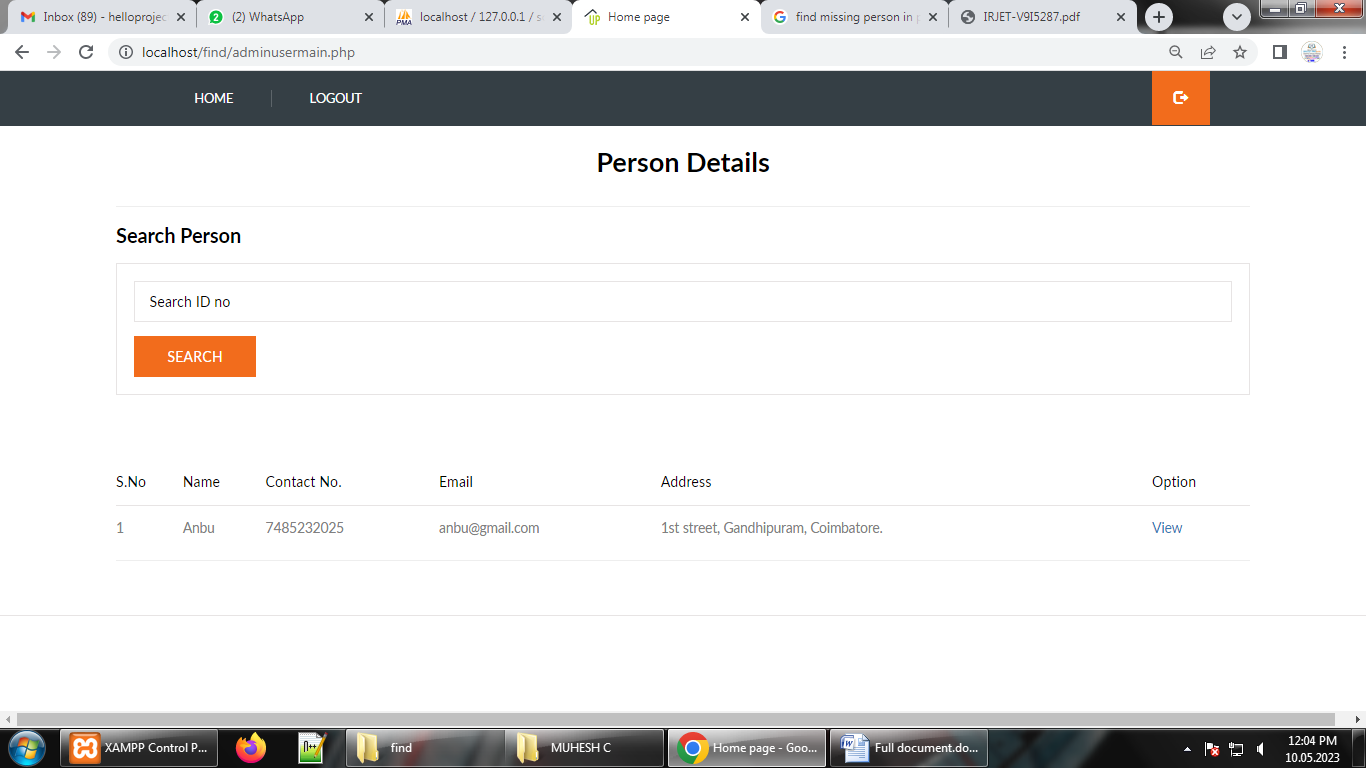
****

**Admin login**

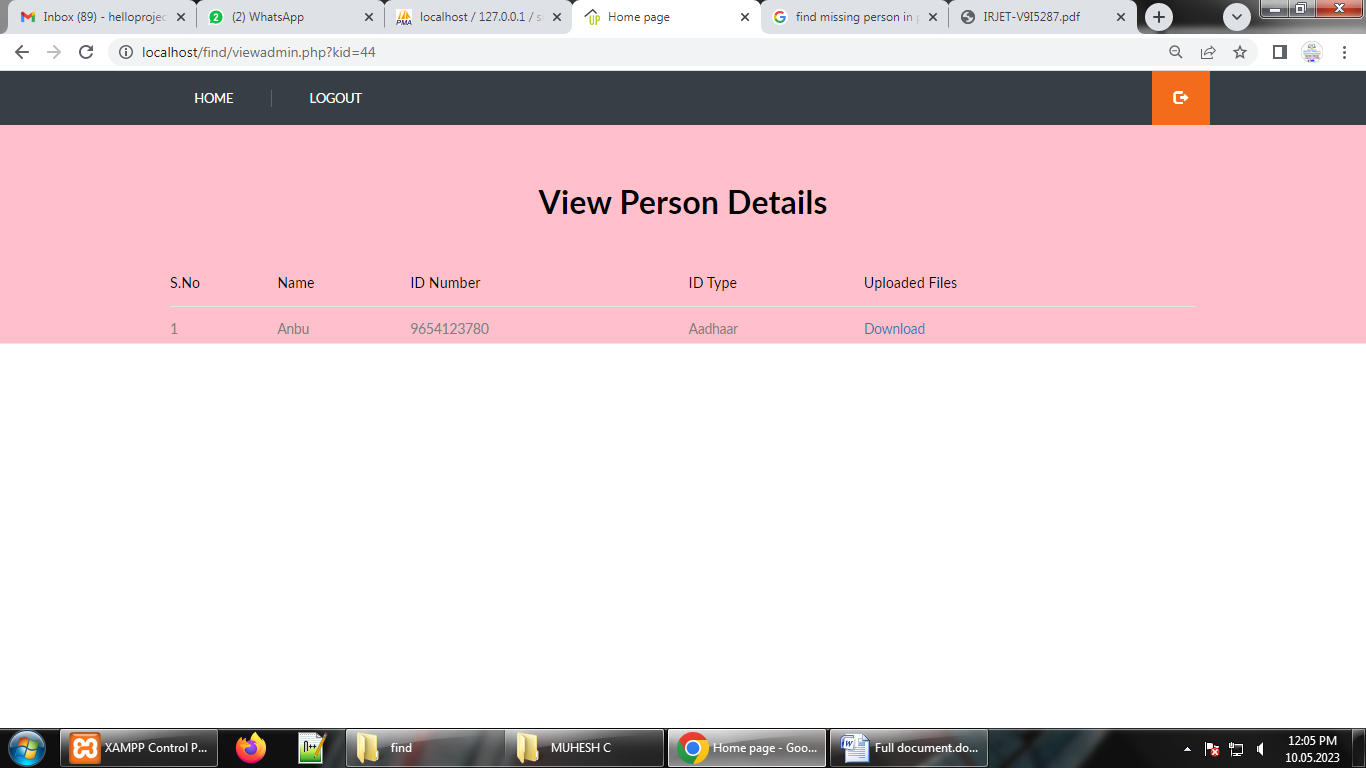
****

****

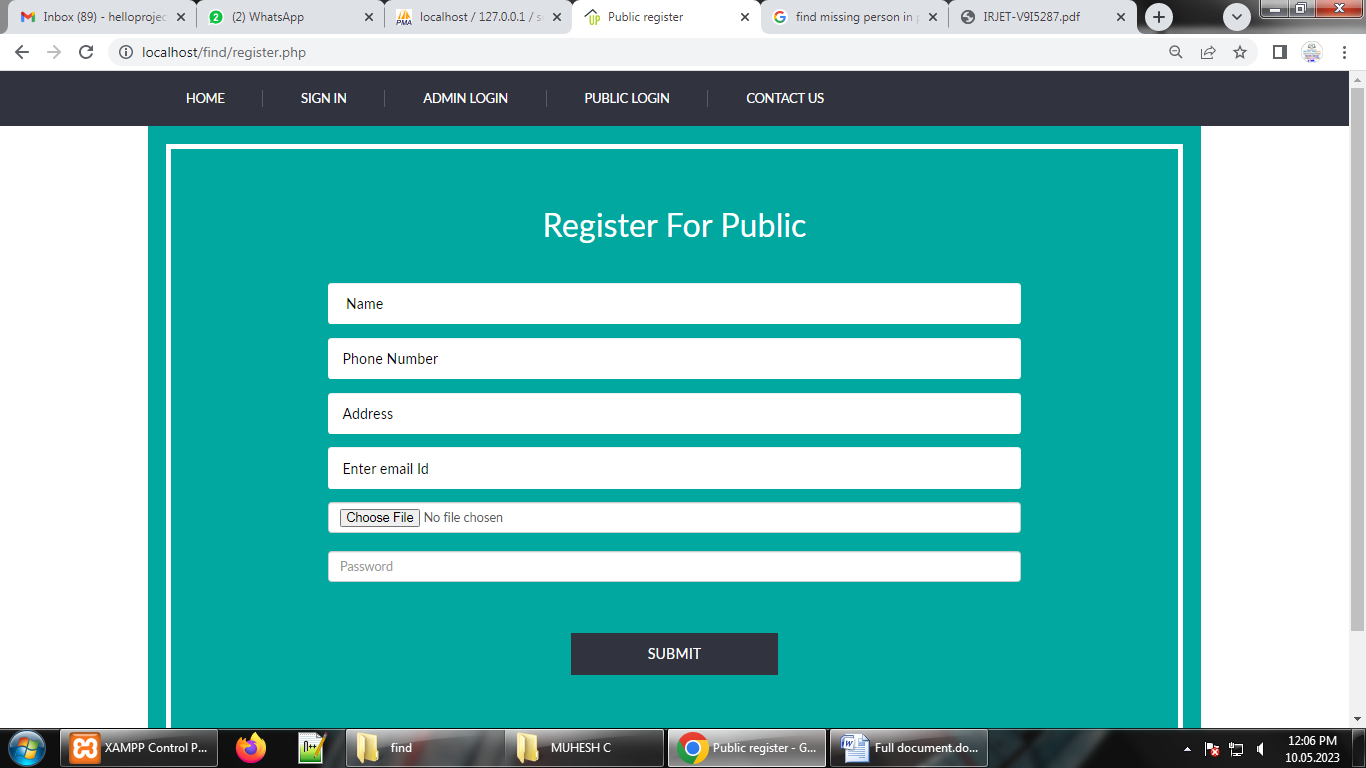
**View public registration**

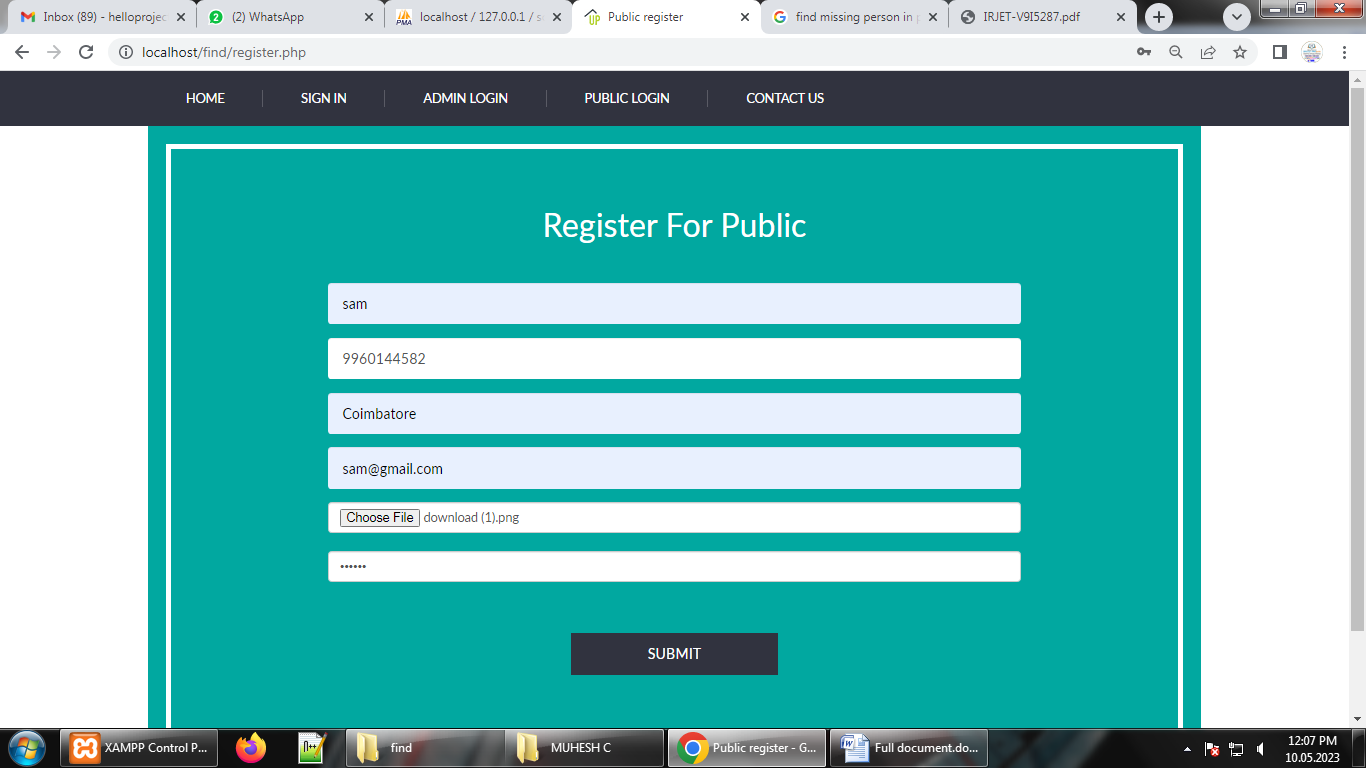
****

**View ID proof details**

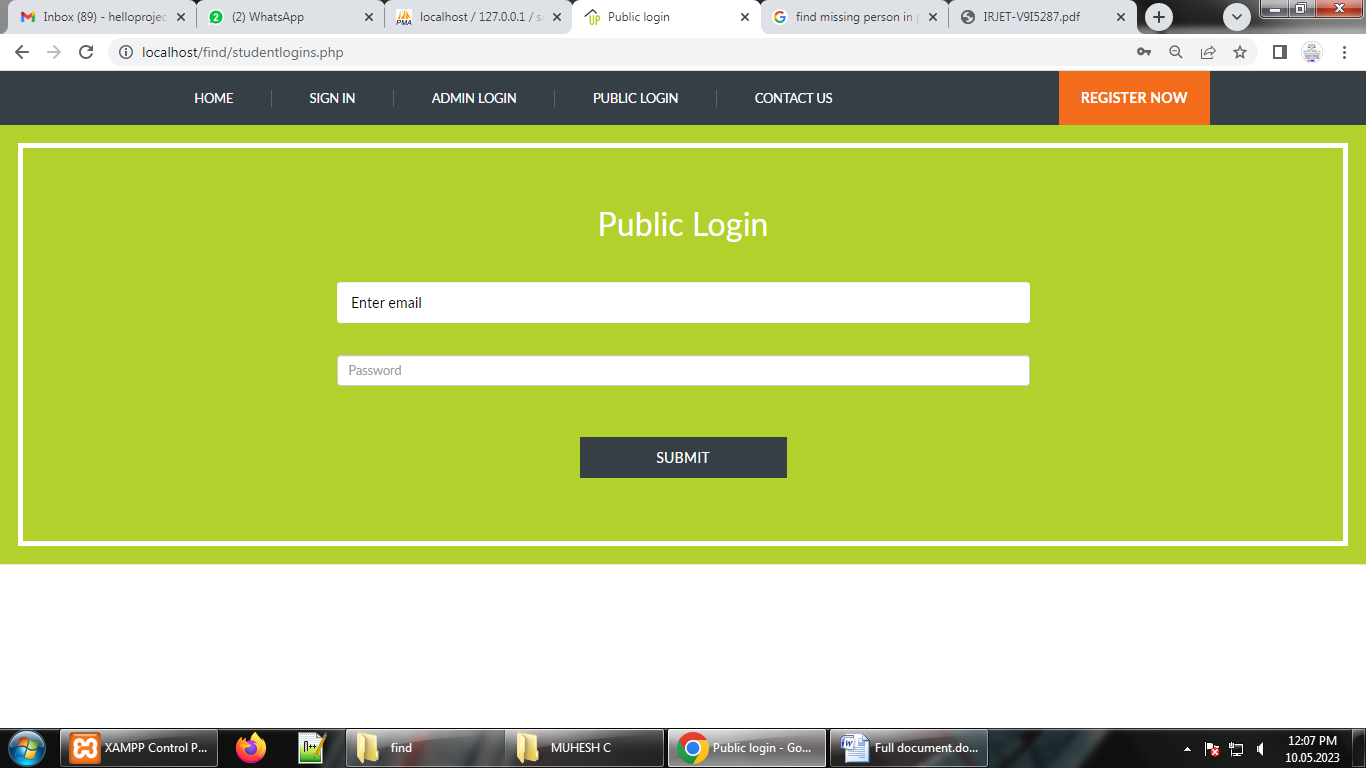
****

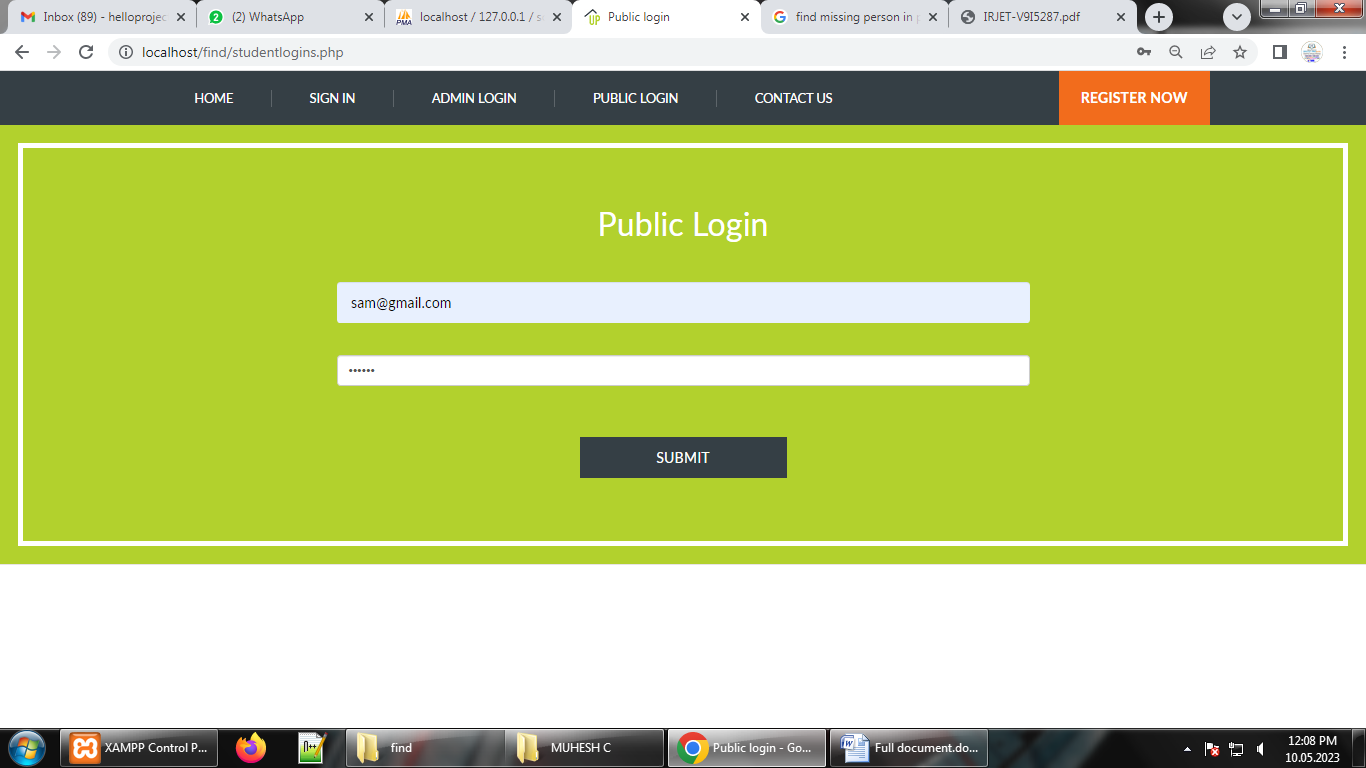
**Public Registration**

****

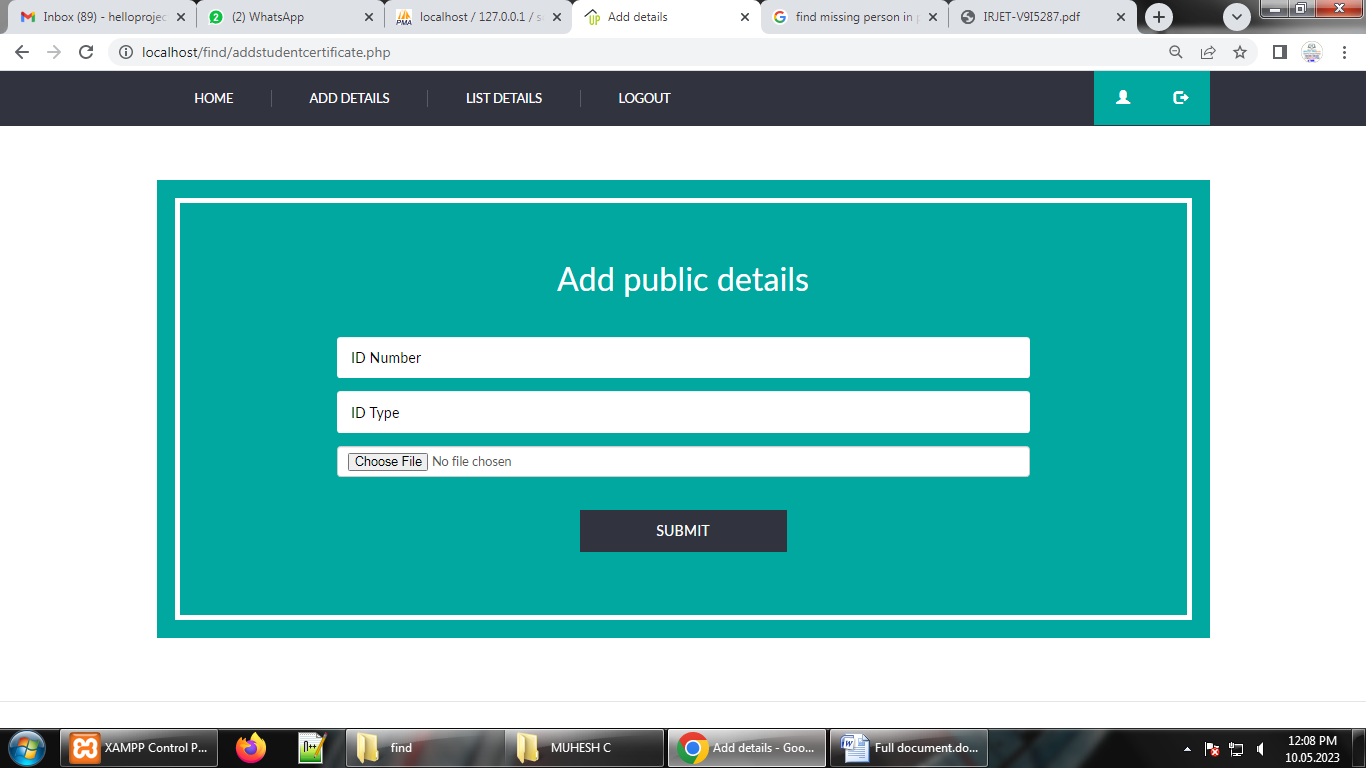
****

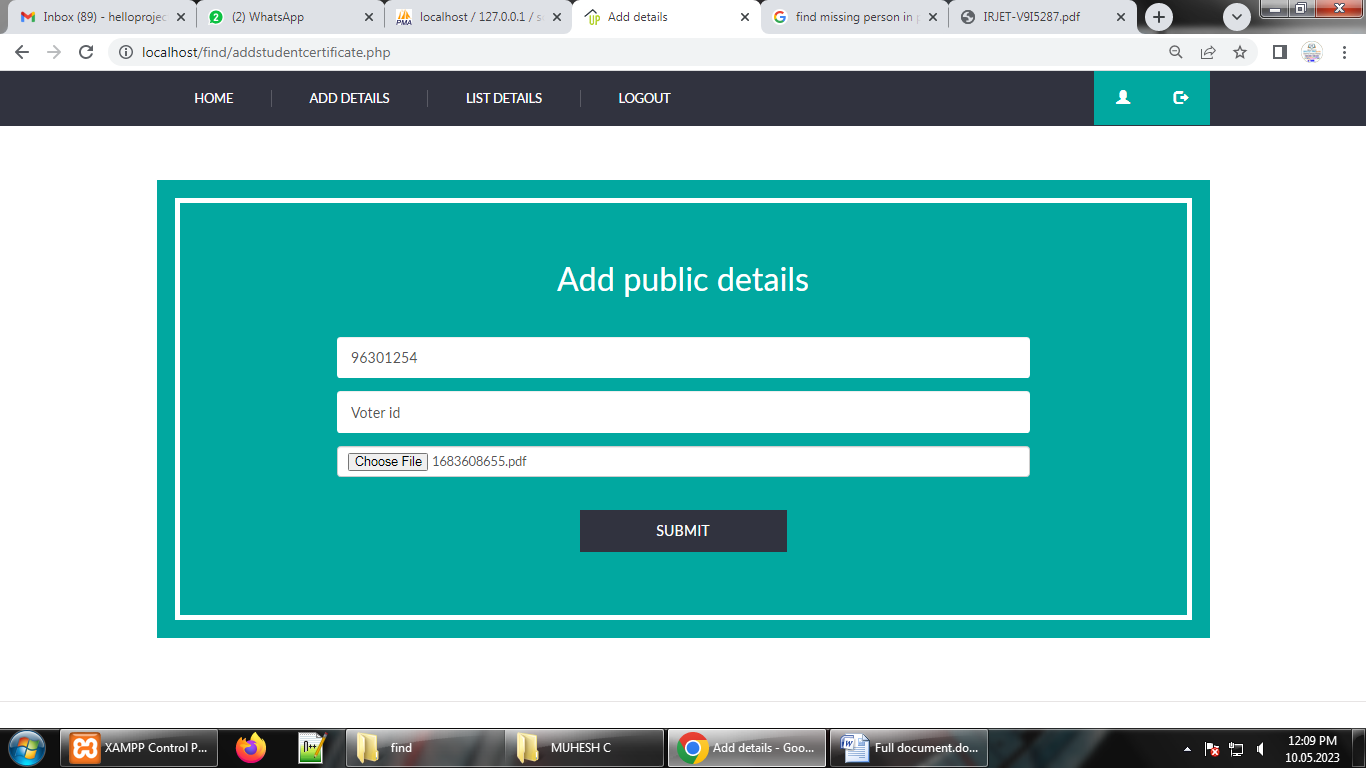
**Public Login**

****

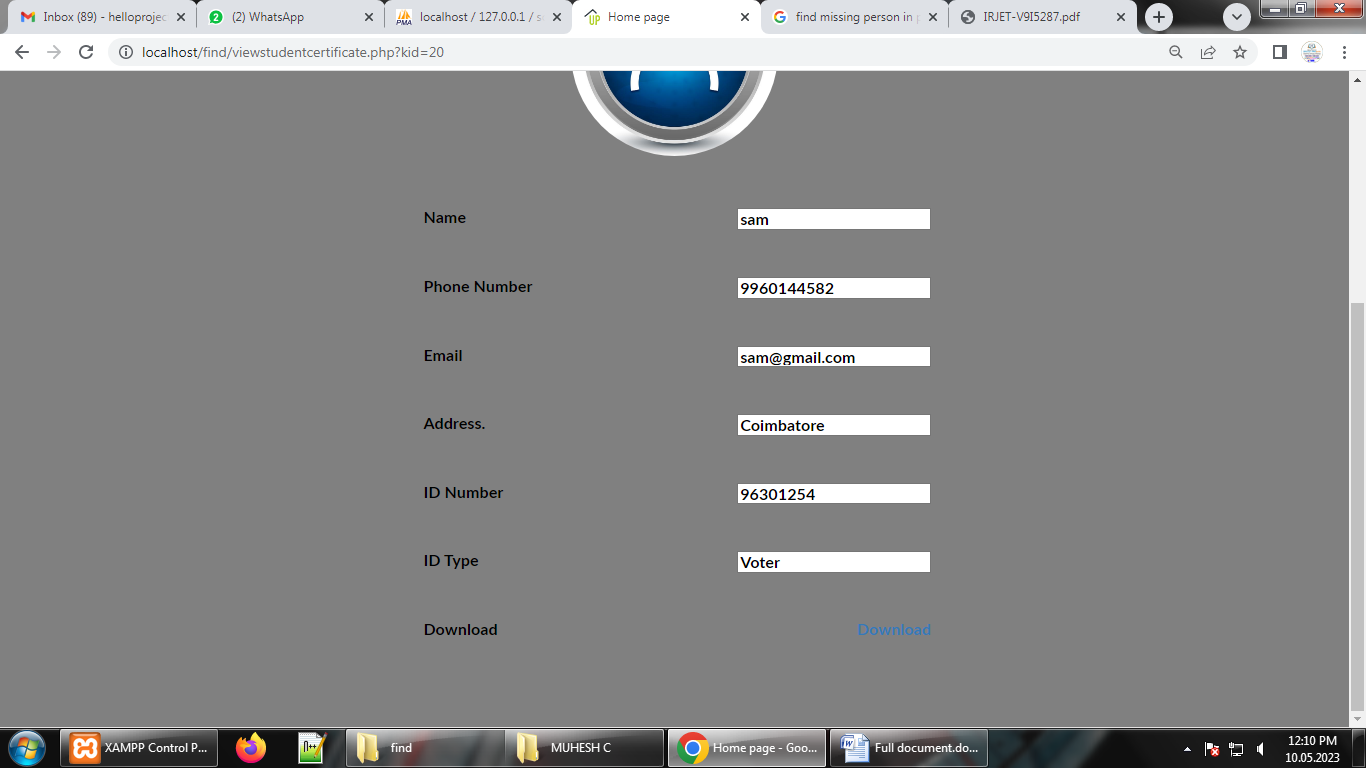
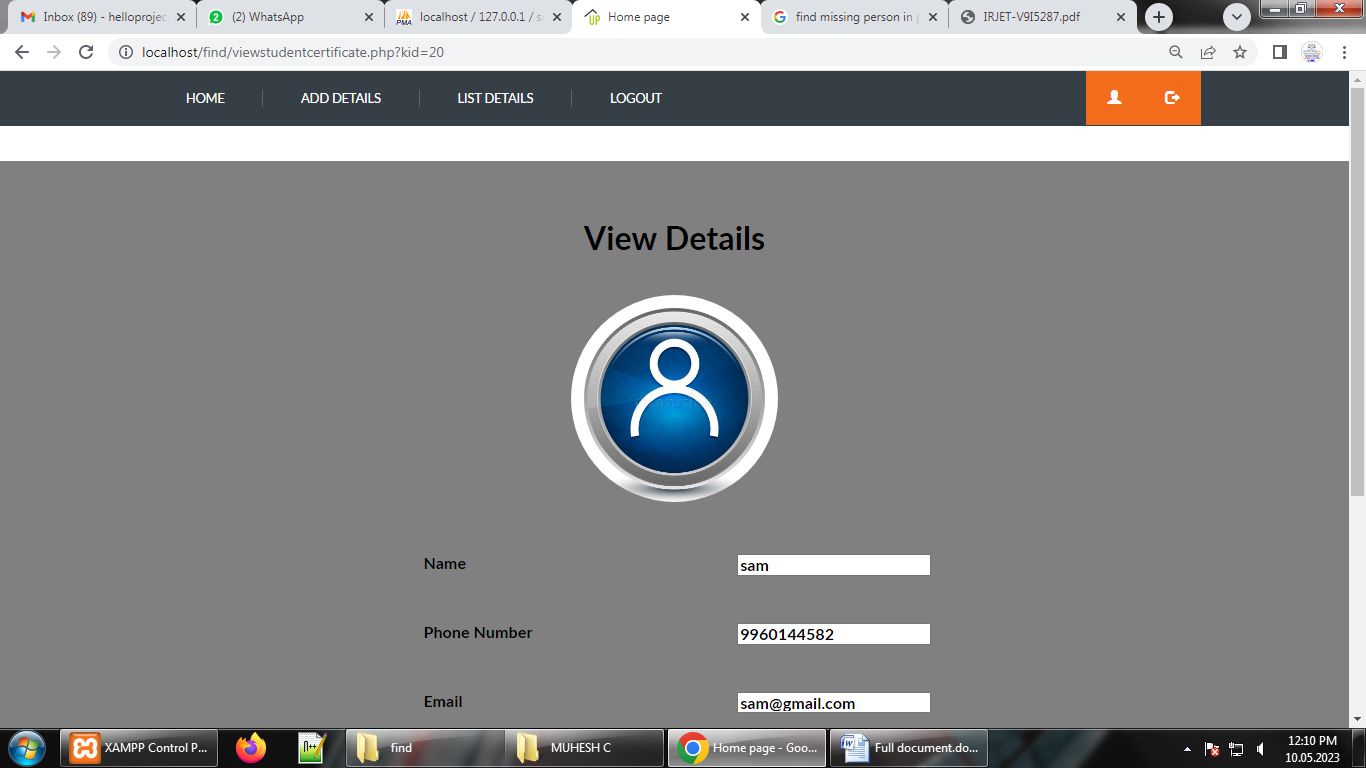
****

**Add ID proof details**

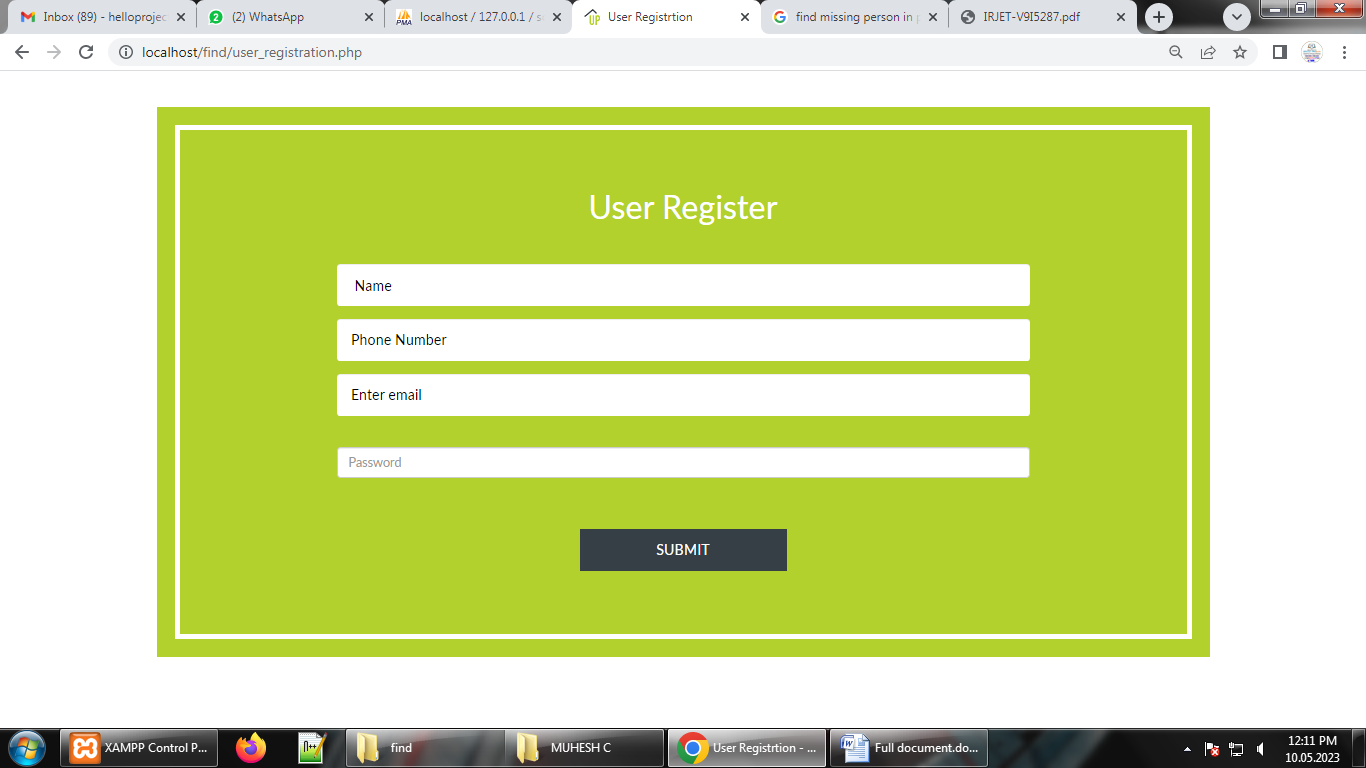
****

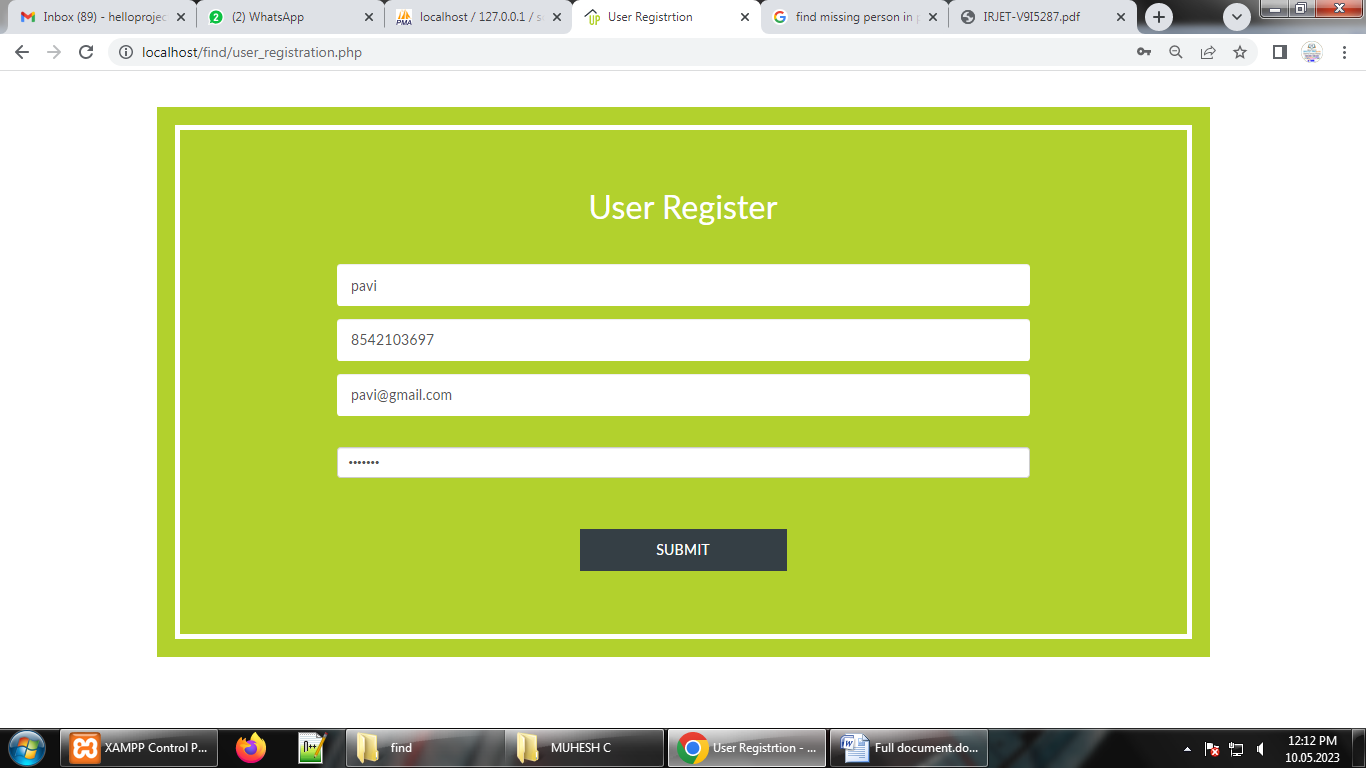
****

**Download certificate**

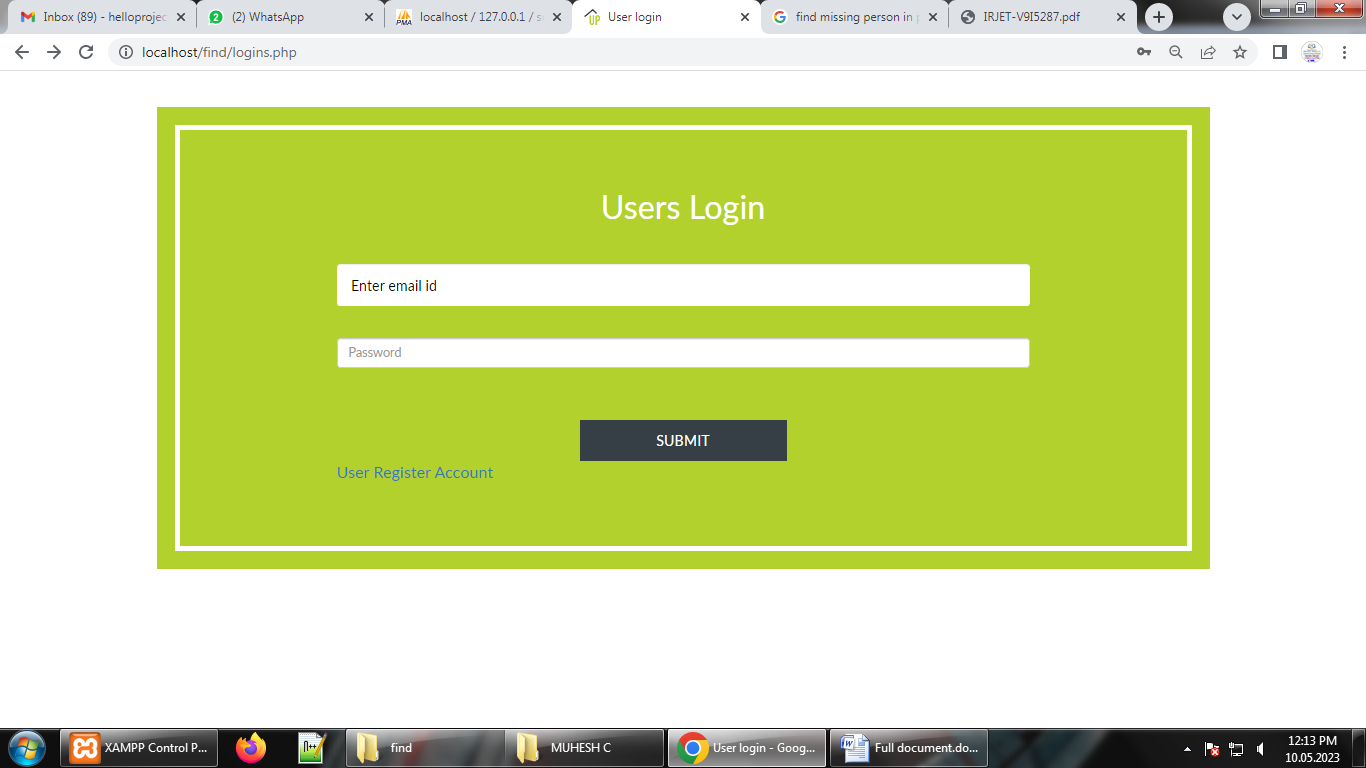
****

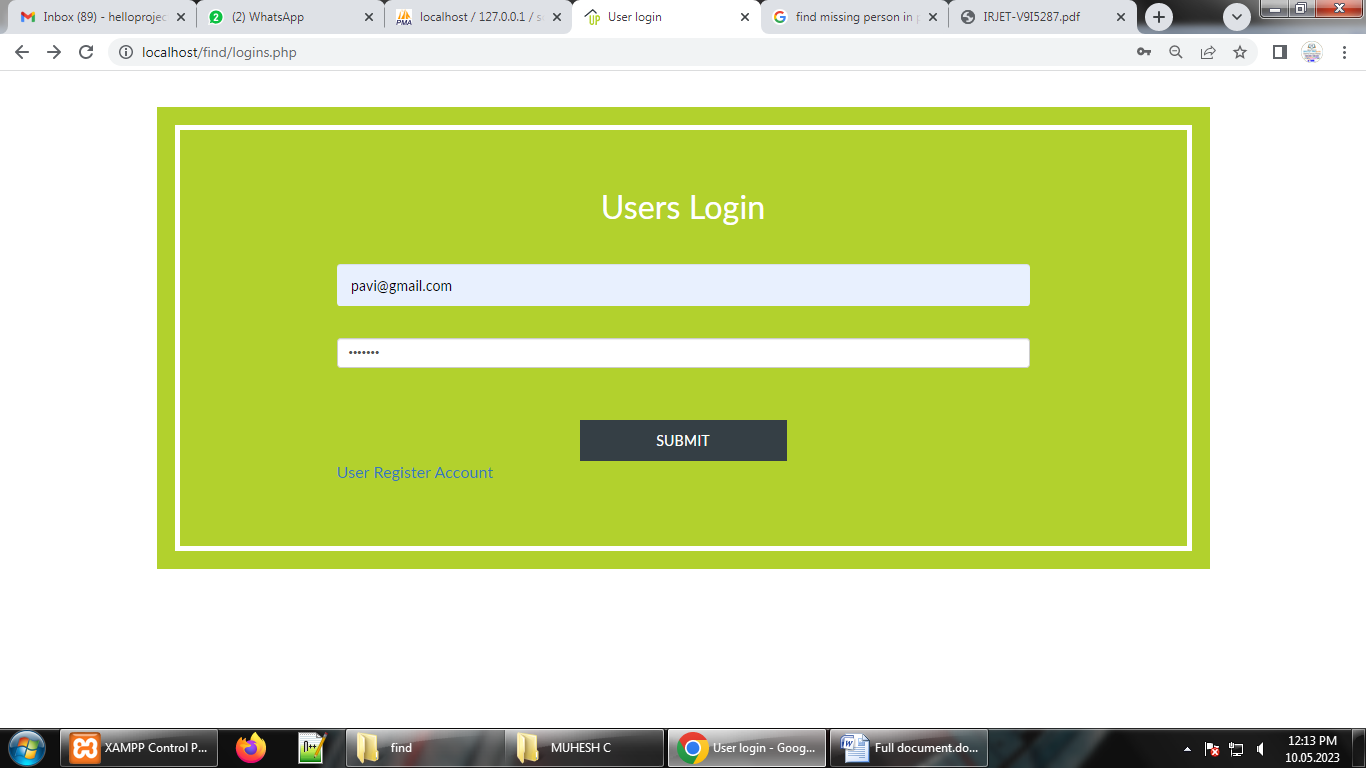
**User registration**



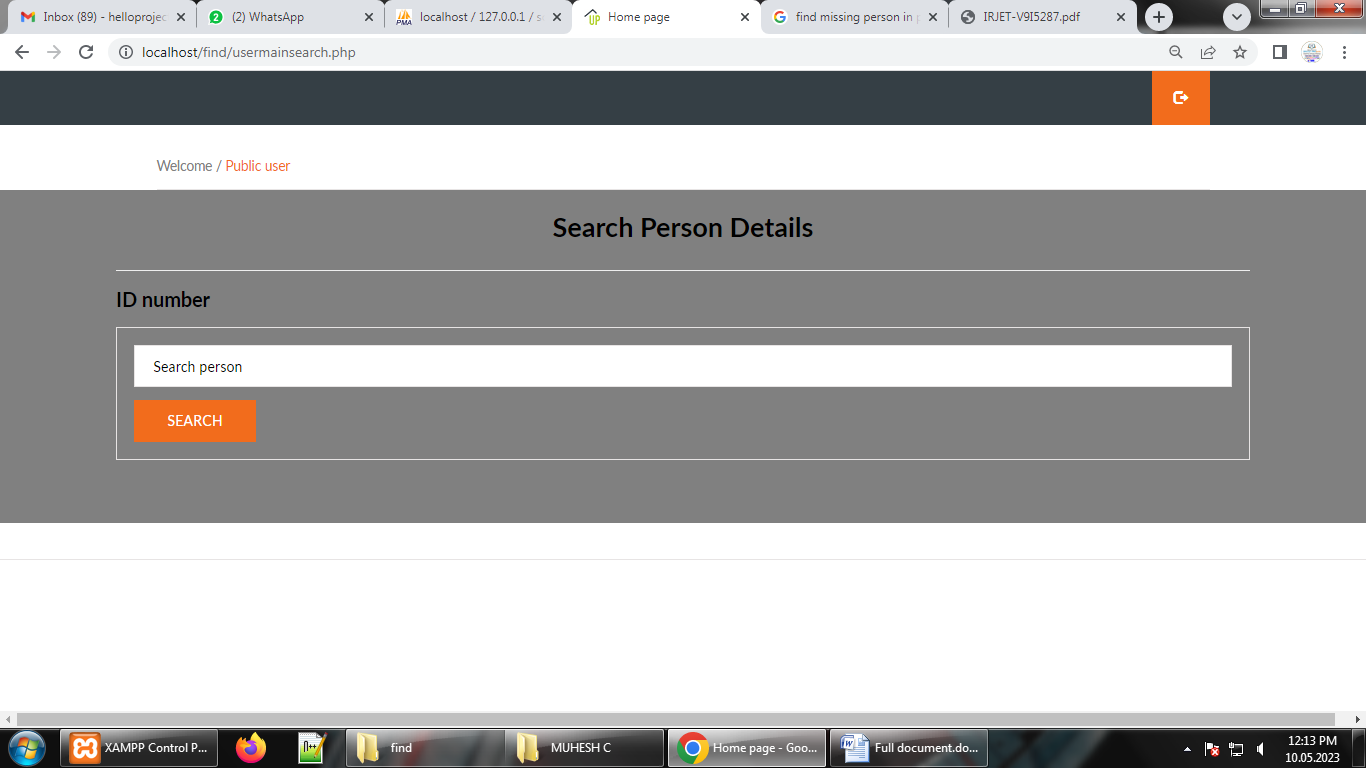
****

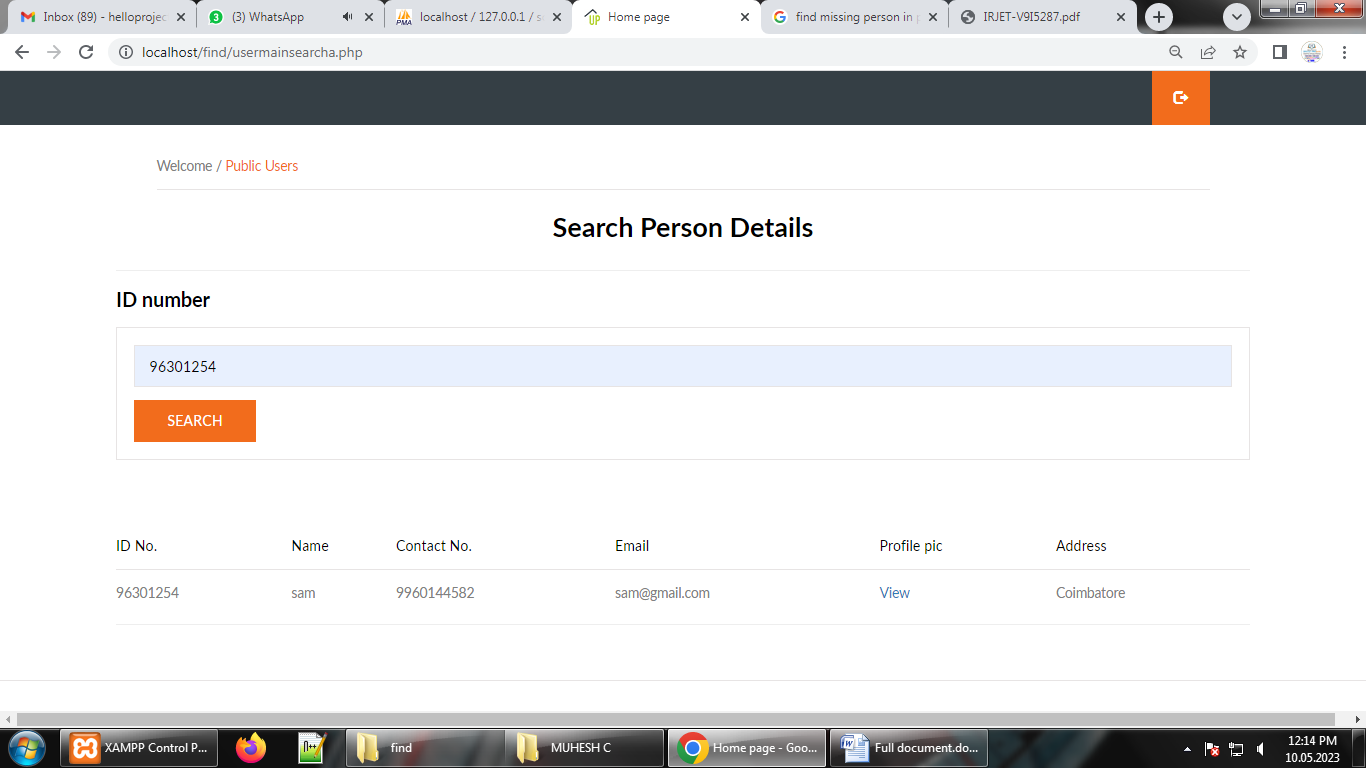
**User Login**

****

****

**Search ID number**

****

****

**E. SAMPLE CODING**

<?php

@ob\_start();

error\_reporting(0);

session\_start();

require('dbconnect.php');

if (isset($\_POST['Alogin'])) {

require('db-admin-login.php');

}

elseif (isset($\_POST['Mlogin'])) {

require('db-member-login.php');

}

else{

}

?>

<!DOCTYPE html>

<html>

<!-- Mirrored from static.pixum.co/up-real-estate-html/index.html by HTTrack Website Copier/3.x [XR&CO'2014], Sat, 30 Mar 2019 05:43:21 GMT -->

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="format-detection" content="telephone=no"/>

<link rel="shortcut icon" type="image/x-icon" href="images/favicon/1.ico"/>

<title>Home page</title>

<link rel="stylesheet" type="text/css" media="screen" href="css/lib/bootstrap/bootstrap.min.css">

<link rel="stylesheet" type="text/css" media="screen" href="fonts/font-awesome-4.4.0/css/font-awesome.min.css">

<link rel="stylesheet" type="text/css" media="screen" href="vendors/jquery-ui-1.11.4/jquery-ui.min.css">

<link rel="stylesheet" type="text/css" media="screen" href="vendors/jcarousel/css/jquery.jcarousel.css">

<link rel="stylesheet" type="text/css" media="screen" href="css/style.css" />

</head>

<body>

<div class="loader">

<div class="loader\_inner"></div>

</div>

<div class="extra-header">

<div class="container">

</div>

<div class="nav-block">

<div class="container">

<div class="row">

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

<?php include("nav.php");?><a href="register.php" class="submit-nav hidden-xs">Register Now</a>

</div>

</div>

</div>

</div>

<div class="mobile-block">

</div>

<div class="home-banner">

<div class="container">

<div class="banner-content">

<div class="banner-message">

<div class="banner-entry">

<span class="entry-title">FIND MISSING PERSON </span>

</div>

</div>

</div>

</div>

</div>

<!--<div id="content" class="container-fluid">

<div class="container">

<div class="row">

<div class="col-sm-8 col-md-9">

<h2 class="block-title ">How to use ... ??</h2><br><br>

<div class="reviews-of-people">

<ul class="reviews-listing">

<li>

<div class="descr">

<span class="title">Register Your Society</span>

<p>Nullam ac vestibulum nisl, in rutrum felis. Pellentesque quis facilisis nisl. Aliquam ut tincidunt sem. Sed at condimentum tellus, vitae elementum odio. Sed hendrerit varius lectus, venenatis blandit ligula pulvinar in. </p>

</div>

</li>

<li>

<div class="descr">

<span class="title">Login with given society code</span>

<p>Nullam ac vestibulum nisl, in rutrum felis. Pellentesque quis facilisis nisl. Aliquam ut tincidunt sem. Sed at condimentum tellus, vitae elementum odio. Sed hendrerit varius lectus, venenatis blandit ligula pulvinar in. </p>

</div>

</li>

<li>

<div class="descr">

<span class="title">Share society code with all society members</span>

<p>Nullam ac vestibulum nisl, in rutrum felis. Pellentesque quis facilisis nisl. Aliquam ut tincidunt sem. Sed at condimentum tellus, vitae elementum odio. Sed hendrerit varius lectus, venenatis blandit ligula pulvinar in. </p>

</div>

</li>

<li>

<div class="descr">

<span class="title">Set committee members</span>

<p>Nullam ac vestibulum nisl, in rutrum felis. Pellentesque quis facilisis nisl. Aliquam ut tincidunt sem. Sed at condimentum tellus, vitae elementum odio. Sed hendrerit varius lectus, venenatis blandit ligula pulvinar in. </p>

</div>

</li>

<li>

<div class="descr">

<span class="title">Set workers</span>

<p>Nullam ac vestibulum nisl, in rutrum felis. Pellentesque quis facilisis nisl. Aliquam ut tincidunt sem. Sed at condimentum tellus, vitae elementum odio. Sed hendrerit varius lectus, venenatis blandit ligula pulvinar in. </p>

</div>

</li>

<li>

<div class="descr">

<span class="title">Maintain Society</span>

<p>Nullam ac vestibulum nisl, in rutrum felis. Pellentesque quis facilisis nisl. Aliquam ut tincidunt sem. Sed at condimentum tellus, vitae elementum odio. Sed hendrerit varius lectus, venenatis blandit ligula pulvinar in. </p>

</div>

</li>

</ul>

</div>

</div>

<div class="col-md-3 col-sm-4" id="login">

<aside class="sidebar main-sidebar">

<div class="widget login">

<br>

<div class="heading">

<span class="widget-title">Login As Member</span>

</div>

<div class="widget-entry gray-bg">

<div class="login-form">

<form method="post" action="login.php" autocomplete="autocomplete">

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

Email:- <br> <br> <input type="email" class="form-control" name="email" required><br>

</div>

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

Password;- <br> <br> <input type="password" class="form-control" name="password" required><br>

</div>

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

<button type="submit" name="submit" class="btn btn-login btn-lg"> Login </button> <br>

</div>

</form>

<a href="forgot-member-pass.php" class="btn-left link">Forgot Password?</a>df

</div><?php

//Include database connection details

//Array to store validation errors

$errmsg\_arr = array();

//Validation error flag

$errflag = false;

if (isset($\_POST['login'])) {

//Function to sanitize values received from the form. Prevents SQL injection

function clean($str) {

$str = @trim($str);

if(get\_magic\_quotes\_gpc()) {

$str = stripslashes($str);

}

return mysqli\_real\_escape\_string($str);

}

//Sanitize the POST values

$emails = ($\_POST['email']);

$passwordm = (($\_POST['password']));

$emailm = mysqli\_real\_escape\_string($db, $emails);

$password = mysqli\_real\_escape\_string($db, $passwordm);

//Input Validations

if($emailm == '') {

$errmsg\_arr[] = 'email ID missing';

$errflag = true;

}

if($password == '') {

$errmsg\_arr[] = 'password missing';

$errflag = true;

}//If there are input validations, redirect back to the username form

if($errflag) {

$\_SESSION['ERRMSG\_ARR'] = $errmsg\_arr;

session\_write\_close();

header("location: index.php");

exit();

}

//Create query

$qry="SELECT \* FROM userregister WHERE email='$emailm' and password='$password'";

$query=mysqli\_query($conn,$qry);

$count = mysqli\_num\_rows($query);

$row = mysqli\_fetch\_assoc($query);

if ($count > 0) {

session\_start();

session\_regenerate\_id();

$\_SESSION['SESS\_MEMBER\_ID'] = $row['user\_id'];

$\_SESSION['SESS\_USER\_NAME']=$row['email'];

$\_SESSION['SESS\_USER\_MOBILE']=$row['mobile'];

$\_SESSION['SESS\_USER\_FNAME']=$row['fname'];

$\_SESSION['SESS\_USER\_LNAME']=$row['lname'];

header('location:usermain.php');

session\_write\_close();

exit();

} else {

session\_write\_close();

?><script language="javascript">

alert("Invalid");

history.back();

</script>

<?php

}

}

?>

</div><br>

<?php

echo $emails;;?>

<div class="heading">

<span class="widget-title">Login As Admin</span>

</div>

<div class="widget-entry gray-bg">

<div class="login-form">

<form method="post" action="index.php" autocomplete="autocomplete">

<input type="number" class="form-control" name="Scode" required="required" placeholder="Society Code \*"><br>

<input type="password" class="form-control" name="password" required="required" placeholder="Password \*"><br>

<?php

if (isset($error))

{

?>

<span style="color: red;"><?php echo $error; ?></span><br><br>

<?php

}

?>

<button class="send-btn" name="Alogin">Login</button>

<a href="forgot-admin-pass.php" class="btn-left link">Forgot Password?</a>

<a href="registration.php" class="btn-right link">Register Now!</a>

</form>

</div>

</div>

</div>

</aside>

</div>

</div>

</div>

</div>

<br><br><br><br>

<div class="row">

<div class="our-features-banner style-3">

<div class="container">

<h2 class="block-title inversed">Our Features</h2>

<span class="sub-title inversed">The best decision for your Society</span>

<div class="row">

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

<div class="single-feature">

<div class="single-feature">

<div class="icon-container">

<div class="icon-border">

<span class="icon lg-icon cash"></span>

</div>

</div>

<span class="main-title">Payment</span>

<span class="featured-sub-title colored">Management</span>

</div>

</div>

</div>

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

<div class="single-feature">

<div class="icon-container">

<div class="icon-border">

<span class="icon lg-icon like"></span>

</div>

</div>

<span class="main-title">Organizing</span>

<span class="featured-sub-title colored">Events</span>

</div>

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

<div class="single-feature">

<div class="icon-container">

<div class="icon-border">

<span class="icon lg-icon deal"></span>

</div>

</div>

<span class="main-title">knew your</span>

<span class="featured-sub-title colored">Neighbours</span>

</div>

</div>

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

<div class="single-feature">

<div class="icon-container">

<div class="icon-border">

<span class="icon lg-icon human"></span>

</div>

</div>

<span class="main-title">SEt Your</span>

<span class="featured-sub-title colored">Profile</span>

</div>

</div>

</div>

</div>

</div>

</div>

<br>

<br>

</div>

<div class="scroll-container">

<div class="container">

<a href="#" class="scroll-top-btn"><i class="fa fa-angle-double-up"></i></a>

</div>

</div>

<div class="agency-container listing with-sidebar">

<div class="row">

<div class="col-md-10 col-md-offset-1 col-sm-12 col-sm-offset-1">

<div class="row">

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

<div class="question-container">

<h4 class="column-title">Comments</h4>

<div class="contacts-block">

<div class="message-form">

<form action="db-add-comments.php" method="post">

<input type="text" name="name" placeholder="Name \*" required="required">

<textarea name="comm" class="message" placeholder="Comment \*" required="required"></textarea>

<button class="send-btn">Comments</button>

</form>

</div></div>

</div>

<?php

$sql = " SELECT \* FROM comment ORDER BY Comment\_Date DESC , Comment\_Time DESC ";

$result = mysqli\_query($conn,$sql);

if (mysqli\_num\_rows($result) > 0)

{

while($row = mysqli\_fetch\_assoc($result))

{?>

<div class="question-container">

<div class="contacts-block">

<div class="message-form">

<div class="top">

<h5><?php echo $row['Name'];?></h5>

</div>

<hr>

<div class="descr">

<p class="descr-text"><?php echo $row['Comment'];?></p>

</div><hr>

<div class="right">

<?php echo $row['Comment\_Date'];?>

<?php echo $row['Comment\_Time'];?>

</div>

</div>

</div>

</div>

<?php

}

}

?>

</div>

</div>

</div>

</div>

</div>-->

<!-- include footer -->

<?php

require('footer.php');

?>

<script type="text/javascript" src="js/libs/jquery.min.js"></script>

<script type="text/javascript" src="js/main.js"></script>

<script type="text/javascript" src="vendors/jquery-ui-1.11.4/jquery-ui.min.js"></script>

<script type="text/javascript" src="vendors/languageswitcher/languageswitcher.js"></script>

<script type="text/javascript" src="vendors/jcarousel/js/jquery.jcarousel.min.js"></script>

<script src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>

</body>

<!-- Mirrored from static.pixum.co/up-real-estate-html/index.html by HTTrack Website Copier/3.x [XR&CO'2014], Sat, 30 Mar 2019 05:44:00 GMT -->

</html>