**TITLE: ROBOTIC PROCESS AUTOMATION TO AUTOMATE CUSTOMER ENQUIRY**

**ABSTRACT**

The biggest management challenge in the new millennium of liberalization and globalization for a business is to serve and maintain good relationship with the customer. In the past suppliers or business took their customers for granted, because at that time the customers were not demanding nor had alternative source of supply or suppliers. But today there is a radical transformation. The changing business environment is characterized by economic liberalization, increasing competition, high consumer choice, demanding customer, more emphasis on quality and value of purchase etc.

All these changes have made today’s producer shift from traditional marketing to modern marketing. Modern marketing calls for more than developing a product, pricing it, promoting it and making it accessible to target customer. It demands building trust, a binding force and value added relationship with the customers.

The process of developing a cooperative and collaborative relationship between the buyer and seller is called customer relationship management shortly called CRM. “The idea of CRM is to know the individual customer intimately, so that the company has a customized product ready for him even before he asks for it.”

**Objectives of CRM**

The CRM is a new technique in marketing where the marketer tries to develop long term relationship with the customers to develop them as life time customers. CRM aims to make the customer climb up the ladder of loyalty. The company first tries to determine who are likely prospects i.e. the people who have a strong potential interest in the product and ability to pay for it.

The company hopes to convert many of its qualified prospect into first time customers and then to convert those first-time customers into repeat customers. Then the company tries to convert these repeat customers into clients – they are those people who buy only from the company in the relevant product categories. The next challenge for the company is to convert those clients into customers. Customers are those clients who praise the company and encourage others to buy from it.

**Modules:**

* Admin Login
* Customer Management
* Customer Enquiry
* Google Calendar Integration
* Event Creation and Notification
* Email Trigger Integration
* Reports

**Admin Login:**

This web application allows only authorized user to manage this app. It needs basic credentials like username and password to authorize the user. This customer management portal verified logon admin and allow them if give details are correct.

**Customer Management:**

For every organization maintain customer details and keep them with up-to-date information is essential for the company. This application will have separate form to add customer details along with their name, contact details and their WhatsApp number and anytime admin can change customer details and they can remove customer details if needed.

**Customer Enquiry:**

Once customer visits the company they came to enquire about company products or services. The details shared with customer and their enquiries will be saved using this module. This enquiry includes enquiry date, customer id, customer name, purpose of enquiry, actual requirement of the customer etc.,

**Google Calendar Integration:**

Keeping customer with company is essential and mandatory task of every organization. So this CRM tool will have the facility to integrate with google calendar of the admin mail id. Google calendar API service provides this feature and the events can be scheduled and integrated with login gmail id.

**Event Creation and Notification:**

Every customer event planned by the admin will be created using this module. Admin can schedule their customer support, customer meeting, services planned with customer etc using module. The event detail includes date of schedule, meeting purpose, notification date and time and these details will be updated with Google calendar and later the notification will be pushed to customer’s mobile or in their gmail.

**Email Trigger Integration:**

Every customer’s were updated through email once their enquired product arrived the store/company. Using this module customer’s get notified through email based on their enquiries, Once the admin includes the arrived product details in the module. The product detail includes product code, product name, quantity, description and the product name will be automatically sent to the customer through email.

**Reports:**

Admin will have detailed report for customer management module, customer enquiries, active and inactive customer details and customer leads and follow-ups of every customer. Admin can view those details at any time.

**Existing System**

The Existing CRM system compiles customer data across different channels, or points of contact, between the customer and the company, which could include the company's website, telephone, live chat, direct mail, marketing materials and [social networks](https://www.techtarget.com/whatis/definition/social-media). The Existing system also gives customer-facing staff members detailed information on customers' personal information, purchase history, buying preferences and concerns. Where, In the existing system, Excel Sheet is used to manually manage the updates given by the customers.

**Disadvantages of the proposed system:**

* Previous system cannot maintain detailed information about customers as every detail maintained manually
* Customer follow up doesn’t have suitable system to monitor their responses
* Admin needs to write down or maintain all customer related activities

**Proposed System**

The Proposed System contains detailed information on customers' personal information, purchase history, buying preferences and concerns. In Which, the details of the customers and their preferences and other details were updated and it is easily maintained in the webpage. Calendar was also included in the proposed system. It is ease accessible from anywhere and remainders were given frequently before the dates. Emails were also sent to the customers automatically based on their enquiries. So, the company doesn’t have the purpose of manually checking for the details and updating to the customers.

**Advantages of the proposed system:**

* Customer management from anywhere is possible by using 24/7 running website
* Complete customer follow up is possible with the help of customer portal
* Google calendar integration have multiple facility and substitutes admin work and maintains good customer lead generation and customer service.
* Adding a mail feature helps to notify the customers automatically, Whenever customer enquired service is completed the automatic mailing service will update customer inbox with services ready to occupy them.

**SYSTEM SPECIFICATION**

**HARDWARE SPECIFICATION**

Processor : Ryzen 5

Display : LG Monitor

Keyboard : 104 Keys

Mouse : Logitech

RAM : 8 GB

Hard Disk : 512 GB

**SOFTWARE SPECIFICATION**

Operating System : Windows 11

Front-End : PHP

Back-End : MySQL

Web Server : APACHE

Browser : IE 6.0 or Later, Firefox, Google Chrome

* 1. **LANGUAGE SPECIFICATION**

**1.3.1. INTRODUCTION TO FRONT END**

**OVERVIEW OF PHP 5.0**

PHP (Hypertext Pre-processor) is a server-side embedded scripting language. Thismeans that it works within an HTML document to confer to it the capacity of generating content ondemand. You can convert your site into a web application, not just a collection of static pages withinformation that may not get updated quite so often, which may be alright for a "personal" web site.

PHP is the 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 very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.

A PHP file may contain text, HTML tags and scripts. Scripts in a PHP file are executed on the server. Of course general scripting or programming languages like Perl, Python, etc. have also platform independence, and are open source.

PHP was designed to work on the web, and in this ambit it excels connecting and querying a database is a simple task that can be handled in 2 or 3 lines of code. The PHP scripting engine is well optimized for the response times needed on web applications it can even be part of the web server itself improving the throughput even more.

The PHP language features the usual complement of control structures, operators, variable types, function declarations and class/object declarations that we have been accustomed to expect from any compiled or interpreted language, and yet it also has features of its own. For example, in C you employ pointers, in other scripting languages this can be cumbersome or even not possible, but in PHP this is just one use of variable variables.

PHP comes with a myriad of options, both to build the distribution and also to configure an installation. PHP supports several APIs and interfaces to other programming tools. Before we get involved in the detail of installing PHP, it would be worthwhile to consider what we would expect to do with PHP in the near future

**Platforms and Web Servers**

PHP is supported on quite a few platforms, many of them UNIX-like and of course on Microsoft’s operating systems supporting the Win32 environment. Since PHP cannot do much without a web server, when we talk of installing PHP on a platform, we also need to take into consideration the web server that we plan to use.

On the UNIX front, Apache is the web server of choice for most installations. PHP is rated as the second most popular Apache module, where it is found to perform very well. PHP can be compiled as a standalone interpreter, like any other CGI scripting language or it can be compiled as an Apache module. Apache is also available on the Windows platform and PHP is supported for this combination.

**FEATURES**

**HTTP authentication with PHP**

The HTTP Authentication hooks in PHP are only available when it is running as an Apache module and is hence not available in the CGI version. In an Apache module PHP script, it is possible to use the header() function to send an "Authentication Required" message to the client browser causing it to pop up a Username/Password input window. Once the user has filled in a username and a password, the URL containing the PHP script will be called again with the predefined variables PHP\_AUTH\_USER, PHP\_AUTH\_PW, and AUTH\_TYPE set to the user name, password and authentication type respectively. These predefined variables are found in the $\_SERVER and $HTTP\_SERVER\_VARS arrays.

Both "Basic" and "Digest" (since PHP 5.1.0) authentication methods are supported. See the header() function for more information. Instead of simply printing out PHP\_AUTH\_USER and PHP\_AUTH\_PW, as done in the above example, you may want to check the username and password for validity. Perhaps by sending a query to a database, or by looking up the user in a dbm file.

**COOKIES**

PHP transparently supports HTTP cookies. Cookies are a mechanism for storing data in the remote browser and thus tracking or identifying return users. You can set cookies using the setcookie() or setrawcookie() function. Cookies are part of the HTTP header, so setcookie() must be called before any output is sent to the browser. This is the same limitation that header() has. You can use the output buffering functions to delay the script output until you have decided whether or not to set any cookies or send any headers.

**SESSION:**

Session support in PHP consists of a way to preserve certain data across subsequent accesses. This enables you to build more customized applications and increase the appeal of your web site. A visitor accessing your web site is assigned a unique id, the so-called session id. This is either stored in a cookie on the user side or is propagated in the URL.

The session support allows you to register arbitrary numbers of variables to be preserved across requests. When a visitor accesses your site, PHP will check automatically (if session.auto\_start is set to 1) or on your request (explicitly through session\_start() or implicitly through session\_register()) whether a specific session id has been sent with the request. If this is the case, the prior saved environment is recreated.

**USING REMOTE FILES**

As long as allow\_url\_fopen is enabled in php.ini, you can use HTTP and FTP URLs with most of the functions that take a filename as a parameter. In addition, URLs can be used with the include(), include\_once(), require() and require\_once() statements (since PHP 5.2.0, allow\_url\_include must be enabled for these). See Supported Protocols and Wrappers for more information about the protocols supported by PHP.

**GARBAGE COLLECTION**

**Reference Counting Basics**

A PHP variable is stored in a container called a "zval". A zval container contains, besides the variable's type and value, two additional bits of information. The first is called "is\_ref" and is a boolean value indicating whether or not the variable is part of a "reference set". With this bit, PHP's engine knows how to differentiate between normal variables and references. Since PHP allows user-land references, as created by the & operator, a zval container also has an internal reference counting mechanism to optimize memory usage. This second piece of additional information, called "refcount", contains how many variable names (also called symbols) point to this one zval container. All symbols are stored in a symbol table, of which there is one per scope. There is a scope for the main script (i.e., the one requested through the browser), as well as one for every function or method.

**Collecting Cycles**

Traditionally, reference counting memory mechanisms, such as that used previously by PHP, fail to address circular reference memory leaks. As of 5.3.0 PHP however implements the synchronous algorithm from the » Concurrent Cycle Collection in Reference Counted Systems paper which addresses that issue.

A full explanation of how the algorithm works would be slightly beyond the scope of this section, but the basics are explained here. First of all, we have to establish a few ground rules. If a refcount is increased, it's still in use and therefore, not garbage. If the refcount is decreased and hits zero, the zval can be freed. This means that garbage cycles can only be created when a refcount argument is decreased to a non-zero value. Secondly, in a garbage cycle, it is possible to discover which parts are garbage by checking whether it is possible to decrease their refcount by one, and then checking which of the zvals have a refcount of zero.

**Performance Considerations**

We have already mentioned in the previous section that simply collecting the possible roots had a very tiny performance impact, but this is when you compare PHP 5.2 against PHP 5.3. Although the recording of possible roots compared to not recording them at all, like in PHP 5.2, is slower, other changes to the PHP runtime in PHP 5.3 prevented this particular performance loss from even showing.

There are two major areas in which performance is affected. The first area is reduced memory usage, and the second area is run-time delay when the garbage collection mechanism performs its memory cleanups. We will look at both of those issues.

**Reduced Memory Usage**

First of all, the whole reason for implementing the garbage collection mechanism is to reduce memory usage by cleaning up circular-referenced variables as soon as the prerequisites are fulfilled. In PHP's implementation, this happens as soon as the root-buffer is full, or when the function gc\_collect\_cycles() is called. In the graph below, we display the memory usage of the script below, in both PHP 5.2 and PHP 5.3, excluding the base memory that PHP itself uses when starting up.

**Back-End Tool with Version**

**MY-SQL 4.0**

MySQL is a full-featured relational database management system. It is very stable and has proven itself over time. MySQL has been in production for over 10 years. MySQL is a multithreaded server. *Multithreaded* means that every time someone establishes a connection with the server, the server program creates a thread or process to handle that client's requests. This makes for an extremely fast server. In effect, every client who connects to a MySQL server gets his or her own thread.

Yet another feature of MySQL is its portability—it has been ported to almost every platform. This means that you don't have to change your main platform to take advantage of MySQL. And if you do want to switch, there is probably a MySQL port for your new platform.

MySQL also has many different application programming interfaces (APIs). They include APIs for Perl, TCL, Python, C/C++, Java (JDBC), and ODBC. So no matter what your company's expertise is, MySQL has a way for you to access it.

MySQL is also very cheap. For an unlicensed, full version of MySQL, the cost is nothing. To license your copy will currently cost you $200. This is an incredible deal, considering what you are getting for your money.

Database systems that provide half the features that MySQL has can cost tens of thousands of dollars. MySQL can do what they do better and for less.

MySQL is a relational database. It uses tables and columns to hold data that can be related by keys. It is well suited for this role. It is also very well suited for various architectures. It can be used in a strictly client/server architecture or as a standalone database.

**The MySQL Supported Types**

MySQL has various data types that support different functions. A data typeis the type of data a column will store. There can be many different data types inside a table, but each column will store its own specific type of information. You can think of a data type as a kind of definition for a column.

A column defined as an integer column will only hold numeric information, whereas a column defined as a CHAR (10) will hold up to 10 alphanumeric characters. These definitions are the key to a quick and efficient database. There are basically three groups of data formats. The first is obviously numeric. Numeric data is data that is a positive or negative number such as 4 or -50.

Numeric data can also be in hexadecimal format (2ee250cc), scientific notation (2X10^23), or a decimal. The second type is character or string format. This format can consist of letters and numbers, whole words, addresses, phone numbers, and generally anything you have to put quotations around.

It consists of everything that doesn't quite fit into either of the other two categories. Some, like dates and times, could be alphanumeric but are stored like numbers. As well as data types, MySQL also provides column modifiers. These modifiers further help define a column's attributes.

They are AUTO\_INCREMENT, UNSIGNED, PRIMARY KEY, NULL, NOT NULL, and BINARY. A more detailed discussion of column modifiers takes place following the coverage of the basic data types.

MySQL runs on many platforms, and binaries are available for most of them. Binaries are the result of compiling the source code. This is by far the easiest way of acquiring MySQL. The alternative is downloading the source code for your platform and then compiling it.

MySQL has many utilities to import as well as export data. It shares some of the common options, but this utility does a little more. It takes the entire database and dumps it into a single text file.

This file contains all the SQL commands needed to recreate your database. It takes the schema and converts it to the proper DDL syntax (**CREATE** statements), and it takes all the data and creates **INSERT** statements out of them. This utility reverse engineers your database.

**DATA FLOW DIAGRAM:**

**LEVEL 0:**

Administrator

Customer

Enquiry

Update Event

View

**LEVEL 1:**

Admin

Calendar

Customer

Cus\_enquiry

Enq\_follow

Customer Tickets

Email

**LEVEL 2:**

Customer

Customer

Cus\_ticket

**INPUT DESIGN**

The input design is the process of entering data to the system. The input design goal is to enter to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operations are minimized.

The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using it.

Input design is the process of converting user-originated inputs to a computer-based format input data are collected and organized into group of similar data. Once identified, appropriate input media are selected for processing.

The input design also determines the user to interact efficiently with the system. Input design is a part of overall system design that requires special attention because it is the common source for data processing error. The goal of designing input data is to make entry easy and free from errors. In this project, we are giving the input as the video frames are to be stored in files.

**OUTPUT DESIGN**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these results for later consultation.

The output form in the system is either screen or hard copies. Output design aims at communicating the results of the processing to the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels.

**DATABASE DESIGN**

Table Name: Customer

Primary Key: cusid

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Cusid | Int | 4 | Customer id |
| Cusname | Varchar | 30 | Customer name |
| Gender | Varchar | 10 | Gender |
| Address | Varchar | 40 | Address |
| Mobile\_no | Varchar | 12 | Mobile number |
| mailId | Varchar | 40 | Mail id |
| password | Varchar | 40 | Password |
| status | Int | 11 | Status |

Table Name: Customer\_Enquiry

Primary Key: enquiry\_id

Foreign Key: cusid

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Enquiry\_id | Int | 4 | Enquiry Id |
| Cusid | Int | 4 | Customer id |
| Enq\_date | Date/Time | 8 | Enquiry date |
| Category | Varchar | 30 | Category |
| Subject | Varchar | 20 | subject |
| Enq\_details | Varchar | 100 | Enquiry Details |

Table Name: Enquiry\_Followup

Primary Key: id

Foreign Key: enq\_id, cusid

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int | 4 | Id |
| Enq\_id | Int | 4 | Enquiry id |
| Cusid | Int | 4 | Customer id |
| Follow\_date | Date/Time | 8 | Follow up date |
| Remarks | Varchar | 100 | Follow up remarks |

Table Name: Event\_Calendar

Primary Key: id

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int | 4 | Id |
| Notify\_date | Date/Time | 8 | Notification date |
| Time\_from | Time | 4 | Notification time |
| Time\_to | Time | 4 | Notification end time |
| Title | Varchar | 30 | Event title |
| Notify\_details | Varchar | 100 | Notification details |

Table Name: Product

Primary Key: id

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int | 11 | Id |
| Pt\_code | Varchar | 20 | Product code |
| Pt\_name | Varchar | 50 | Product Name |
| Quantity | Int | 4 | Quantity |
| Description | Varchar | 50 | Description |

**4. TESTING AND IMPLEMENTATION**

**4.1. System Testing**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

**The objective of testing is as follows:**

* + Testing is the process of executing a program with the intent of finding an error.
  + A successful test is that one of the cover of undiscovered error.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**Testing Methodologies:**

System testing is the state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together. System testing requires a test plan that consists of several key activities and steps for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system.

Testing is an important stage in software development. The system test in implementation stage in software development. The system test in implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification. Testing is vital to the achievement of the system goals. The objective of testing is to discover errors. To fulfill this objective a series of test step unit, integration, validations and system tests were planned and executed. The test steps are,

**4.2. TYPES OF TESING**

**4.2.1. REGRESSION TESTING:**

Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases that are re-executed to ensure existing functionalities work fine. This testing is done to ensure that new code changes do not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.

**4.2.2. LOAD TESTING:**

When your software development project is nearing completion, there’s one test that’s essential to understanding its readiness for deployment: load testing. This type of [performance testing](https://www.microfocus.com/en-us/what-is/performance-testing) allows you to determine how your web application will behave during normal and peak load conditions, as well as its breaking point (should it occur below the peak load condition). At its core, load testing is used to confirm that your web application meets your intended performance goals or objectives, which are frequently identified in a service level agreement (SLA).

**4.2.3. 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.

**4.2.4. RECOVERY TESTING**

Recovery testing is non -functional testing that determines the capability of the software to recover from failures such as software/hardware crashes or any network failures. Software failures are unavoidable, some failures do not let the complete system down, but some failures can be a disaster. To reduce the impact of the disaster, “Recovery testing” comes in.

**4.2.4. MIGRATION TESTING**

Data migration is the process of moving data from one location to another, one format to another, or one application to another. Generally, this is the result of introducing a new system or location for the data. The business driver is usually an application migration or consolidation in which legacy systems are replaced or augmented by new applications that will share the same dataset.

The migrations are often started as companies move from on-premises infrastructure and applications to [cloud-based storage](https://qatestlab.com/solutions/by-focus-area/cloud-solutions/) and applications to optimize or transform their business. Hence, adequate testing is a high-priority effort to ensure data migration is successful. In this article, we are going to cover the particularities of quality engineering in the data migration process.

**4.3. System Implementation**

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

**The stage consists of**

**System Installation**

This chapter describes the installation procedures to be followed in order to install the specification in the server.

**Installation Pre-requisites**

The following are tools to be installed for deploying the application.

* XAMPP
* Dreamweaver5

The following steps should be followed to deploy the application:

* Create a virtual directory under the directory www, which is created in the drive specified while installing XAMPP.
* All the pages (.php, .html) developed are saved under this directory.
* The users of the application can access the system by specifying the URL in the Web browser.

**System Implementation**

Implementation is used here to mean the process of converting a new or revised system design into operational one; conversion is one aspect of implementation. The other aspect is post implementation review and software and maintenance.

There are three types of implementation:

* Implementation of a computer system
* Implementation of new computer system
* Implementation of a modified application

**Implementation of the computer system**

It’s should be replace a manual system the problem encountered are converting files, training users, creating accurate files, and verifying printouts for integrity.

**Implementation of new computer system**

It’s should replace an existing one this is usually a difficult conversion. If not properly planned there can be many problems. Some large computer system have taken even years to convert.

**Implementation of a modified application**

It’s should be replace an existing one using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes to the file.

**Implementation Procedures**

The implementation phase is less creative than system design. A system project may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase.

The final report to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation.

Several procedures of documents are unique to the conversion phase**.** They includethefollowing,

* The conversion portion of the implementation plan is finalized and approved.
* Files are converted
* Parallel processing between the existing and the new system are logged on a special form.
* Assuming no problems, parallel processing is discontinued. Implementation results are documented for reference.

**CONCLUSION**

From this project it can be concluded that the customer relationship management in Company is satisfaction among customers. The company is using various CRM practices like customization of the product, maintaining interaction with the customers regularly and providing good quality product etc. Customer relationship management has a certain impact on the profitability of the company. Average sale per customer can be increased 15% compared with models. Customer response rate towards marketing activities is also improving. There are various factors affecting the customer relationship management like working environment of the company, support from top management and coordination among the departments of the company. Information technology is not used as much as it should be.

**SCOPE FOR FUTURE ENHANCEMENT**

Proposed model developed with customer management, customer enquiries and customer follow up with innovative Google calendar integration and automatic email triggering method. Notification for customer follow up and customer queries can be done using google calendar. Automatic service remainder and stock update is integrated in this application. Whenever customer enquired service is completed the automatic mailing service will update customer inbox with services ready to occupy them. In future customer enquiry with their product or service queries should be managed and the response can be done from this online. SMS notification can be implemented in necessary area for immediate customer attention.

**BIBLIOGRAPHY**

**Reference books:**

**1) Rasmus Lerdorf, Kevin Tatroe, Peter MacIntyre**

O'Reilly Publications

**2) Web Database Applications with PHP & MySQL, 2nd Edition**

Hugh E. Williams, David Lane

O'Reilly Publications

**3) PHP and MySQL Web Development (4th Edition)**

Luke Welling, Laura Thomson, Addison-Wesley Professional;

O'Reilly Publications

**4) Beginning PHP and MySQL: From Novice to Professional, Third Edition**

(Beginning from Novice to Professional)

Apress; 3rd edition

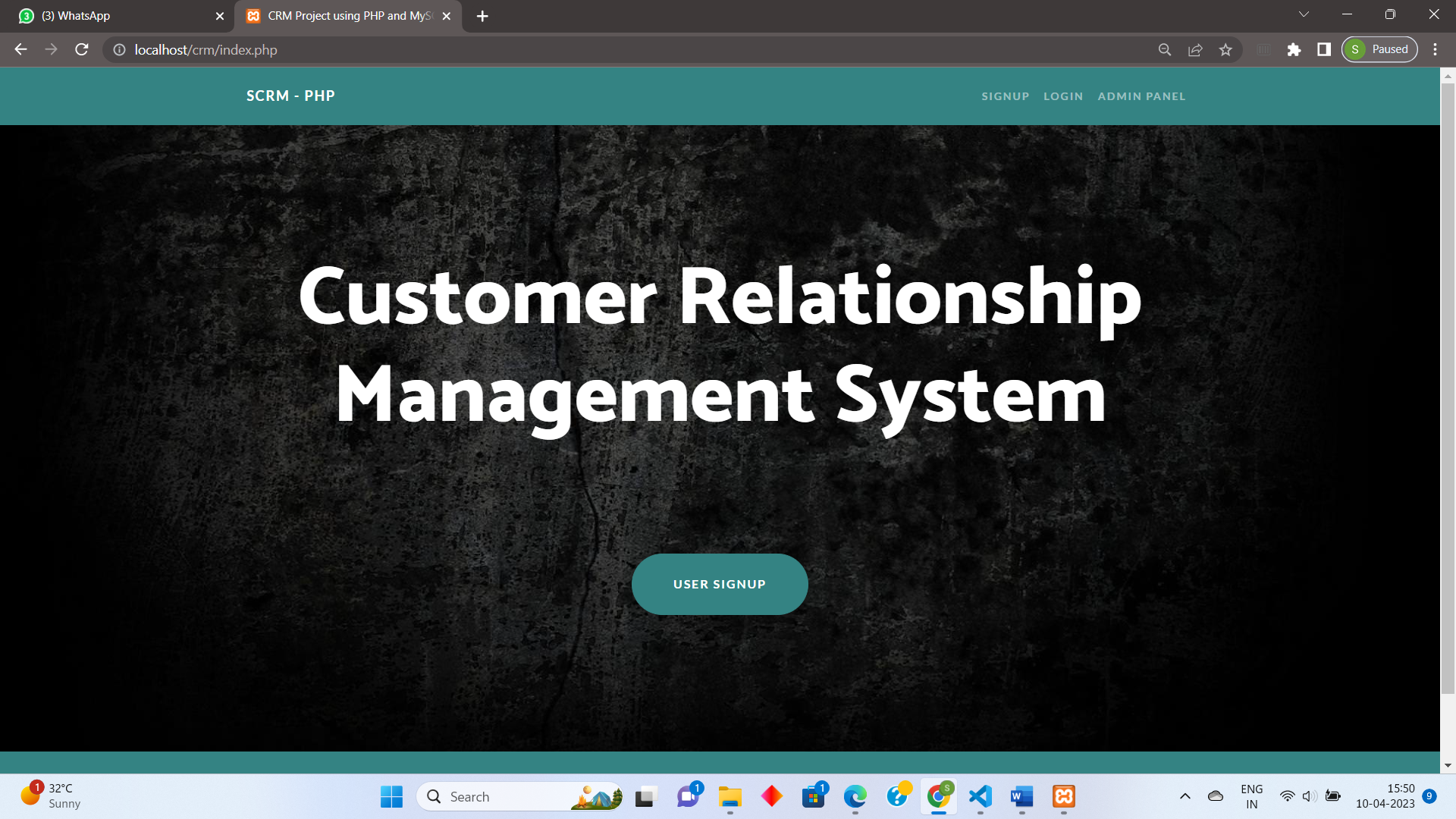
**5) PHP Solutions: Dynamic Web Design Made Easy**

David Powers, friends of ED; illustrated edition

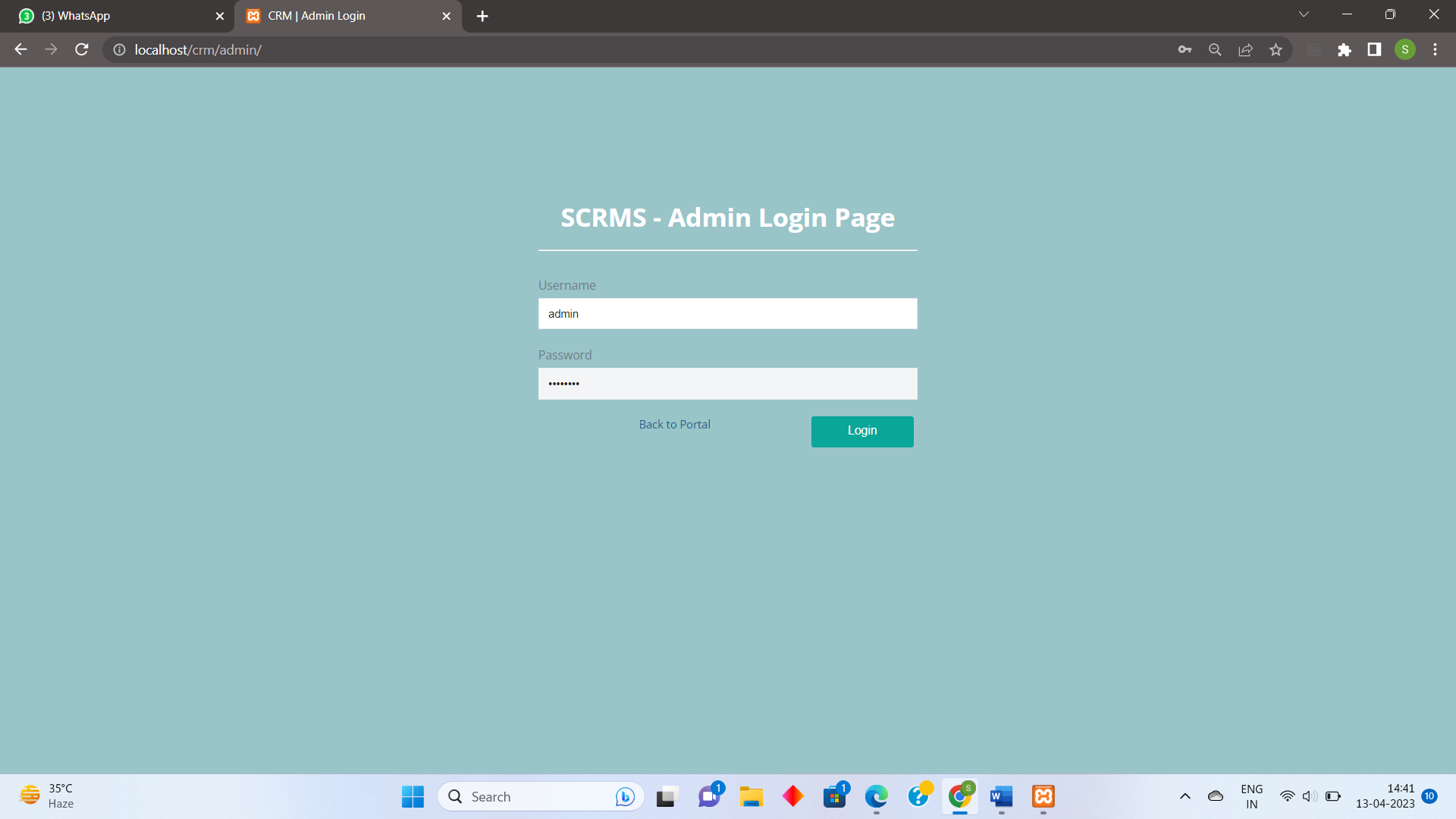
**REFERENCE URL:**

* <http://www.w3schools.com>
* <http://in.php.net/>
* <http://www.php.net/>
* <http://www.php-scripts.com/>
* <http://www.phpro.org/>
* <http://www.codango.com/php>

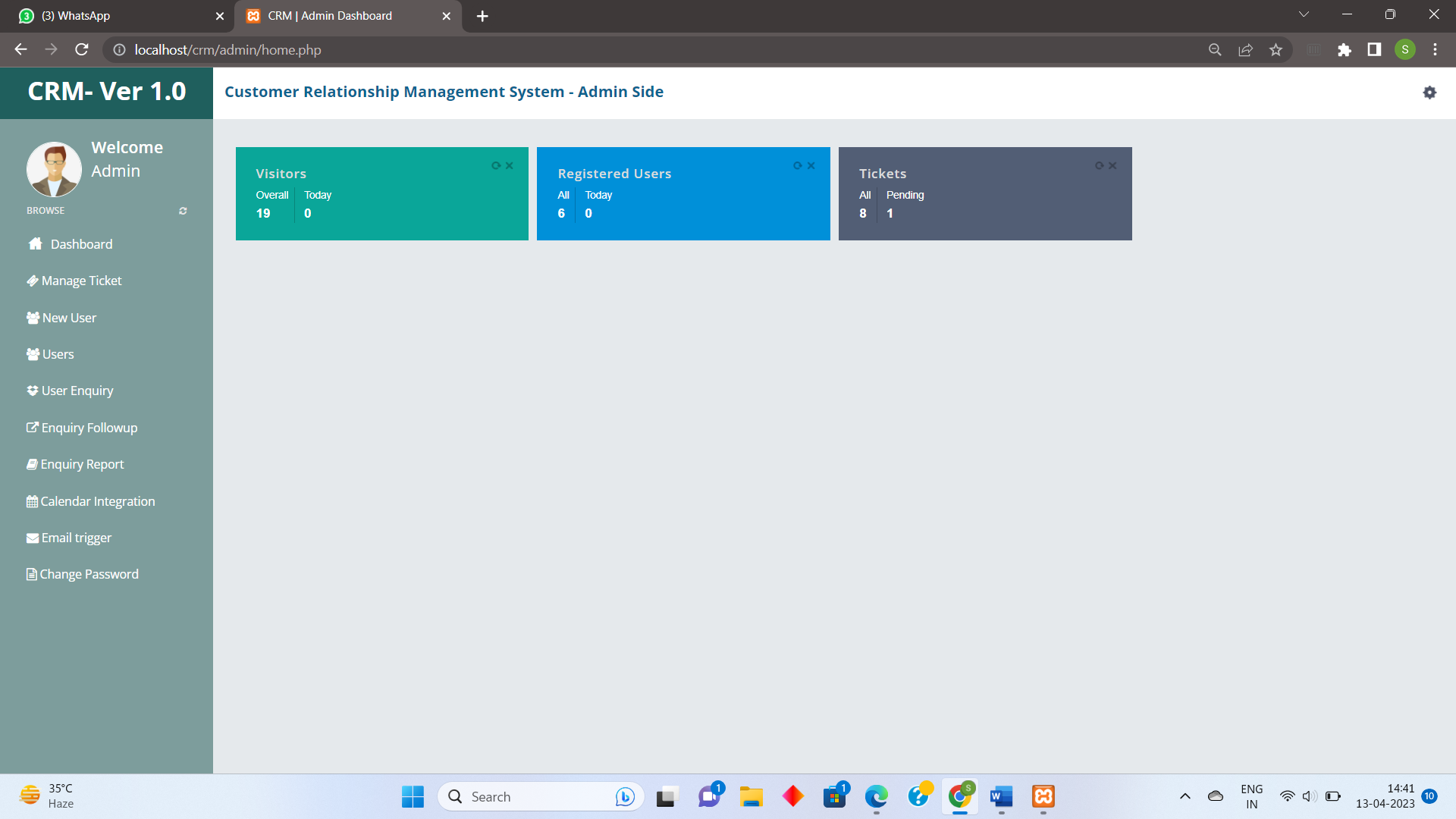
**SCREENSHOT**



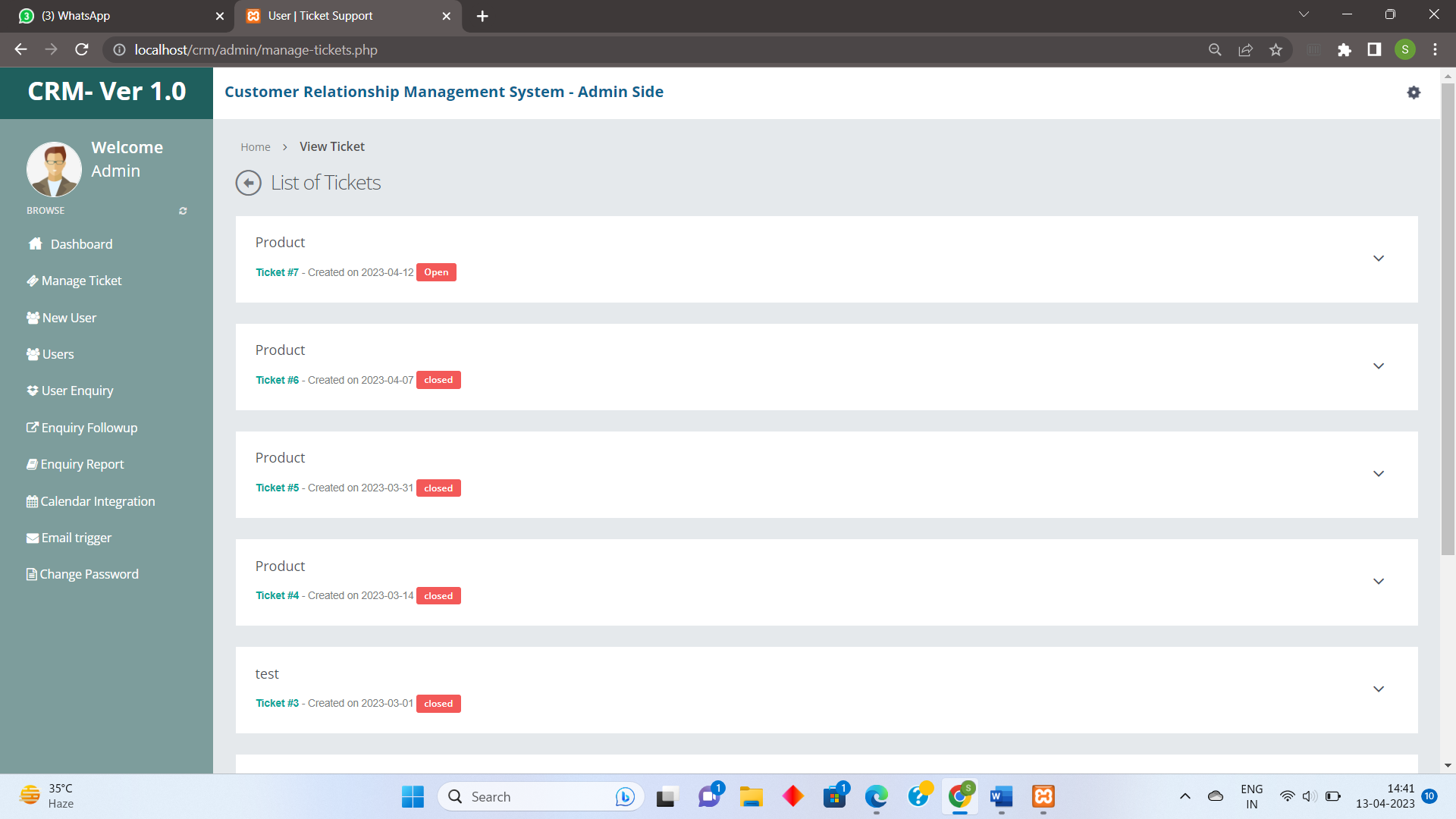
Admin login



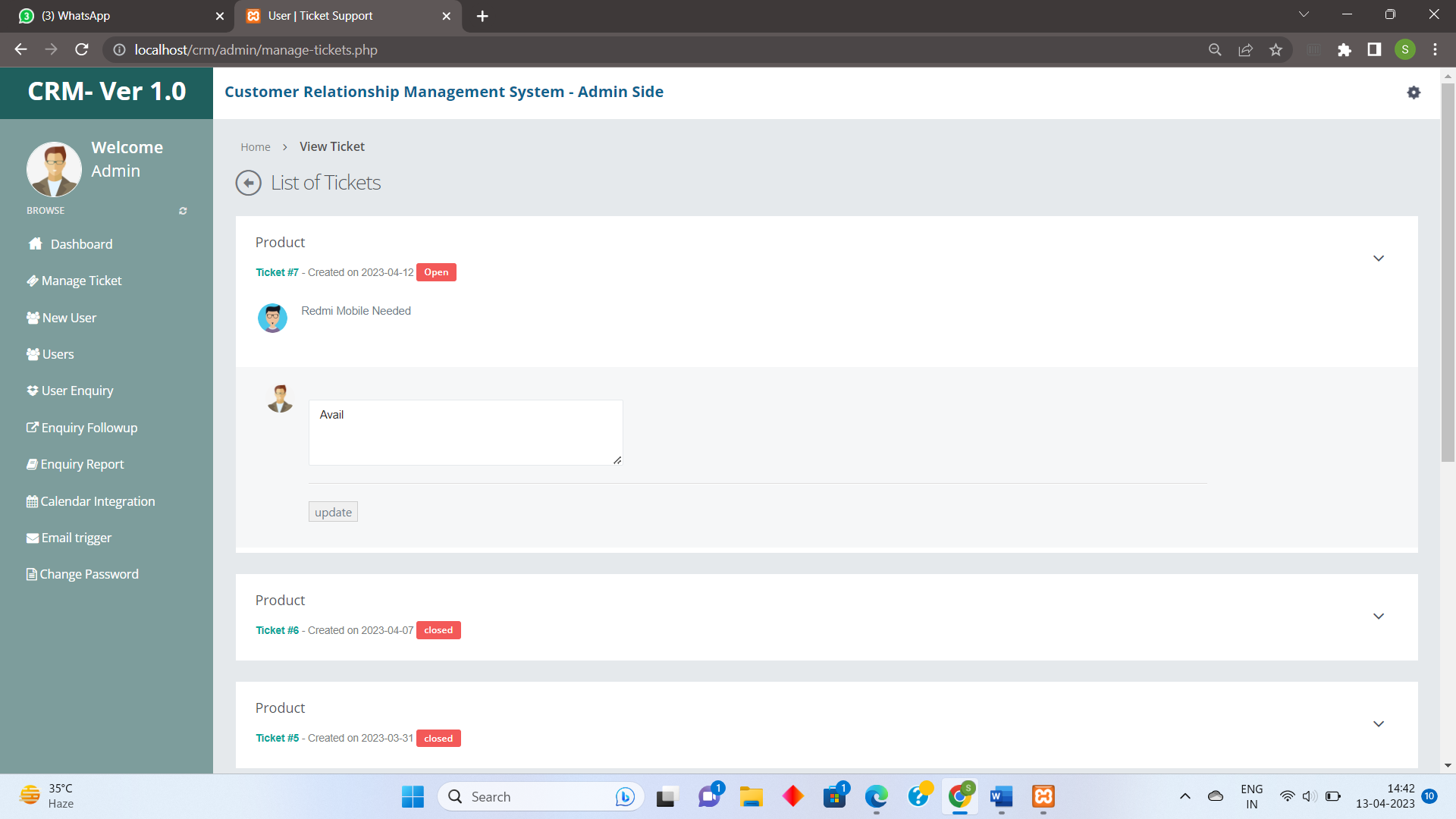
Admin dashboard



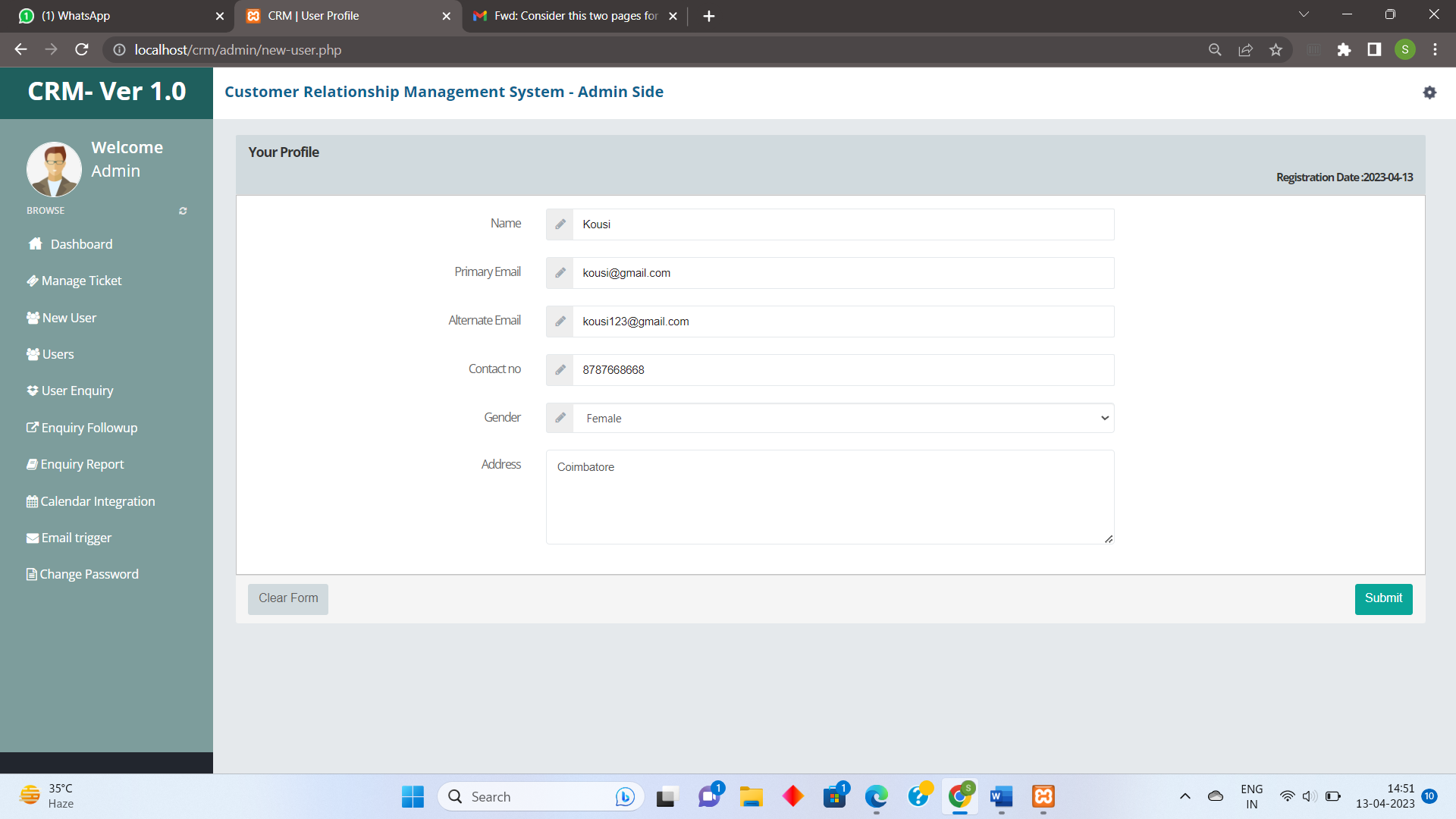
Manage ticket



View Ticket:

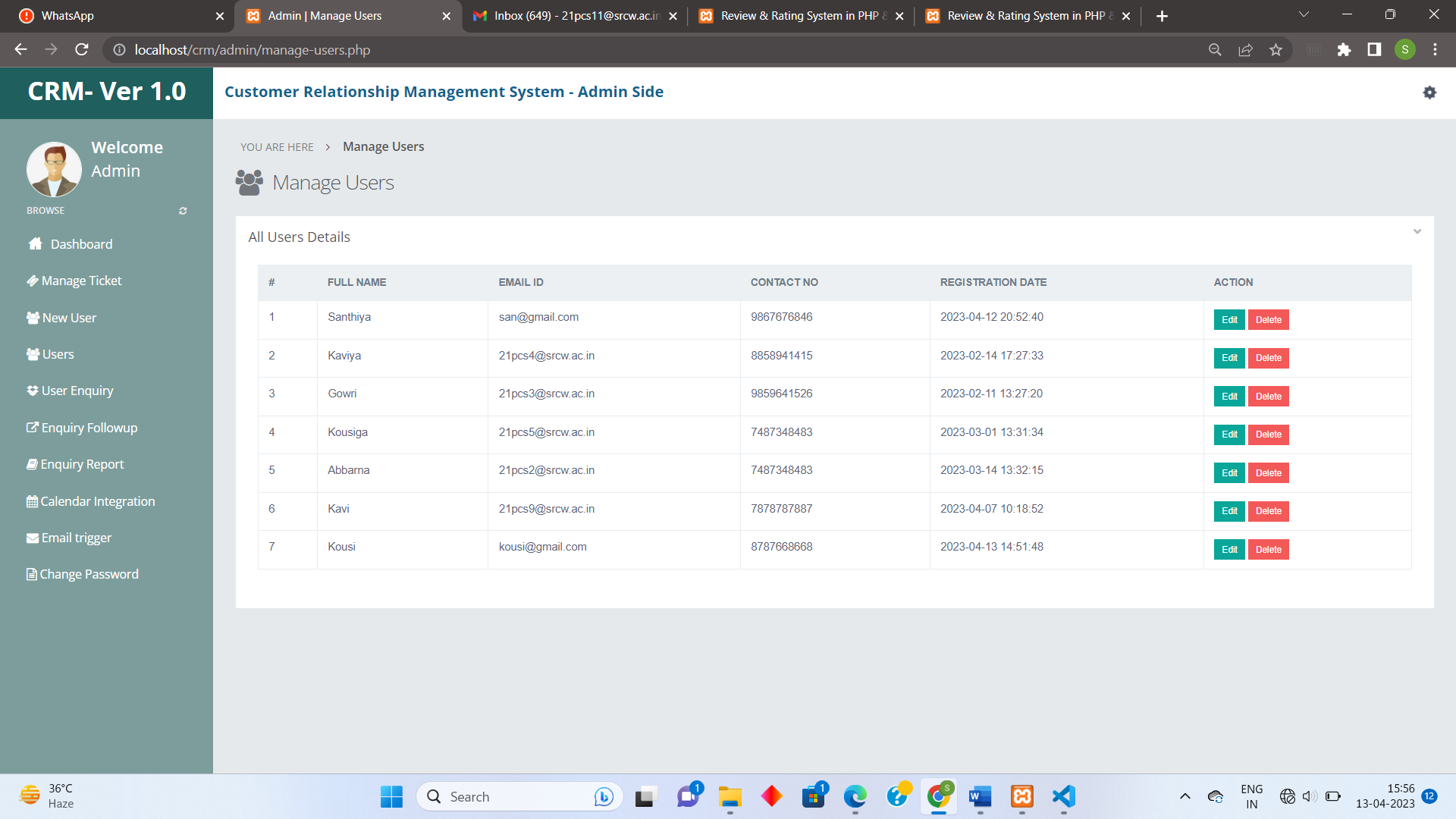


New user:

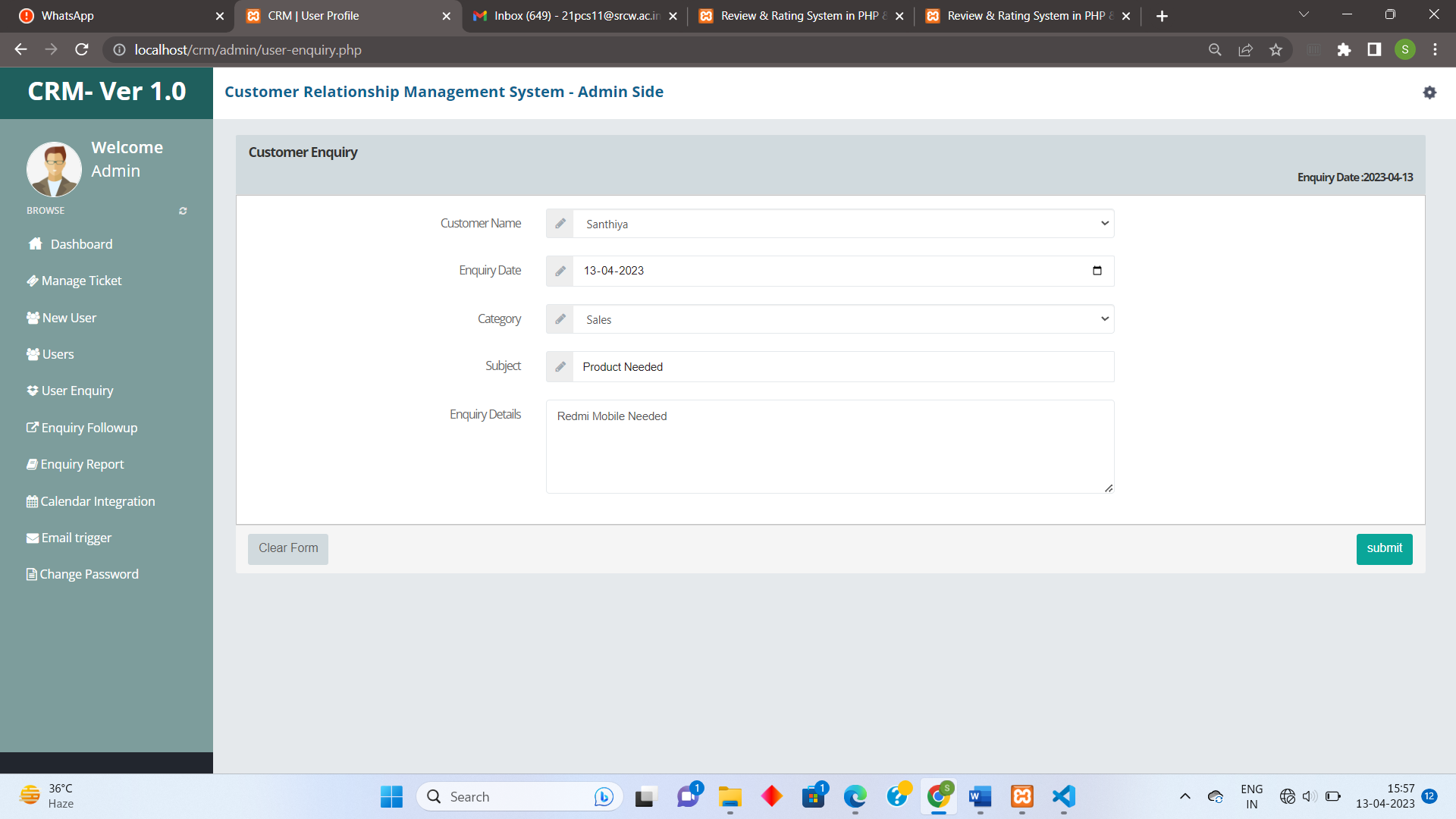


Users:

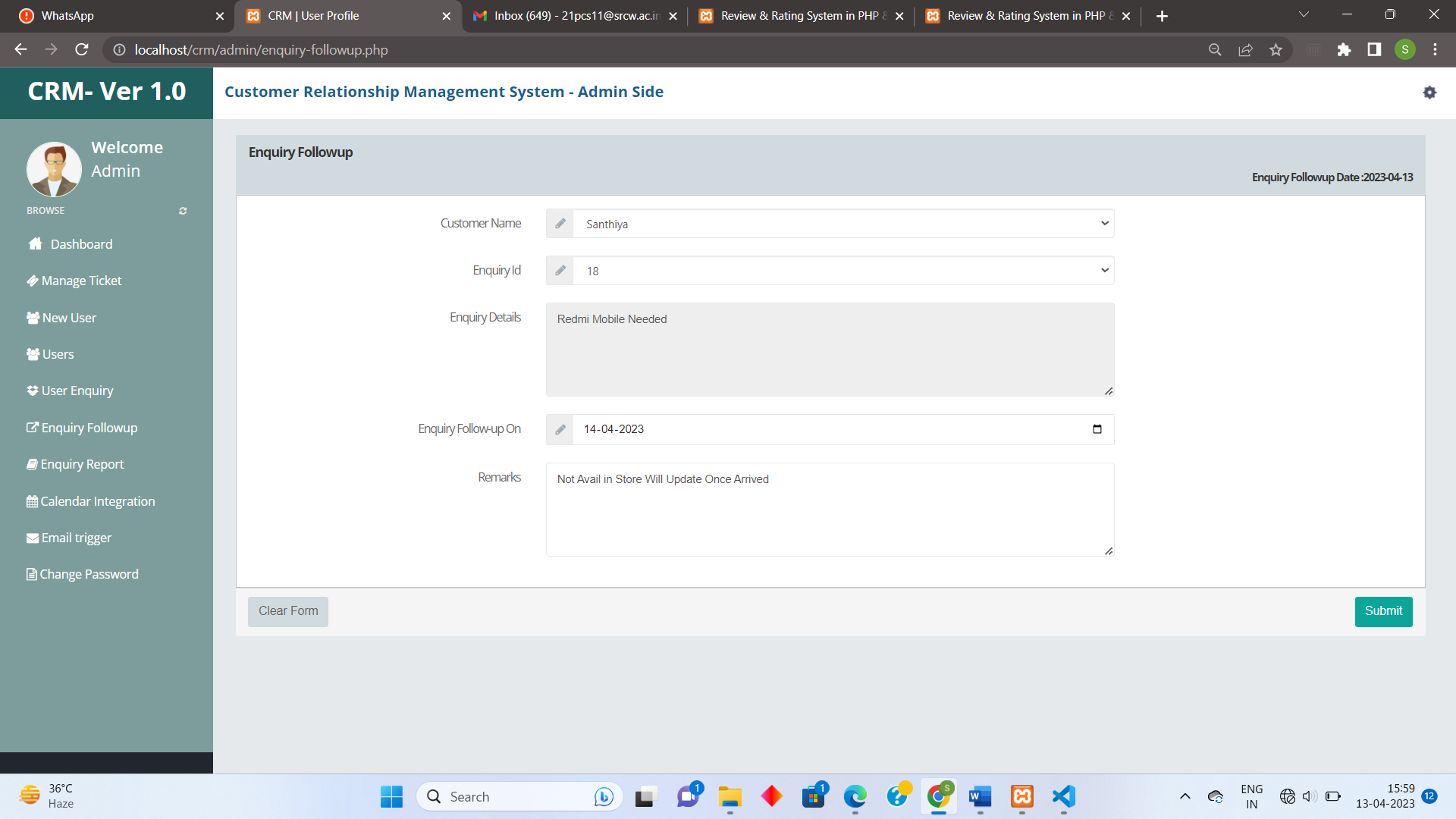
Manage user:



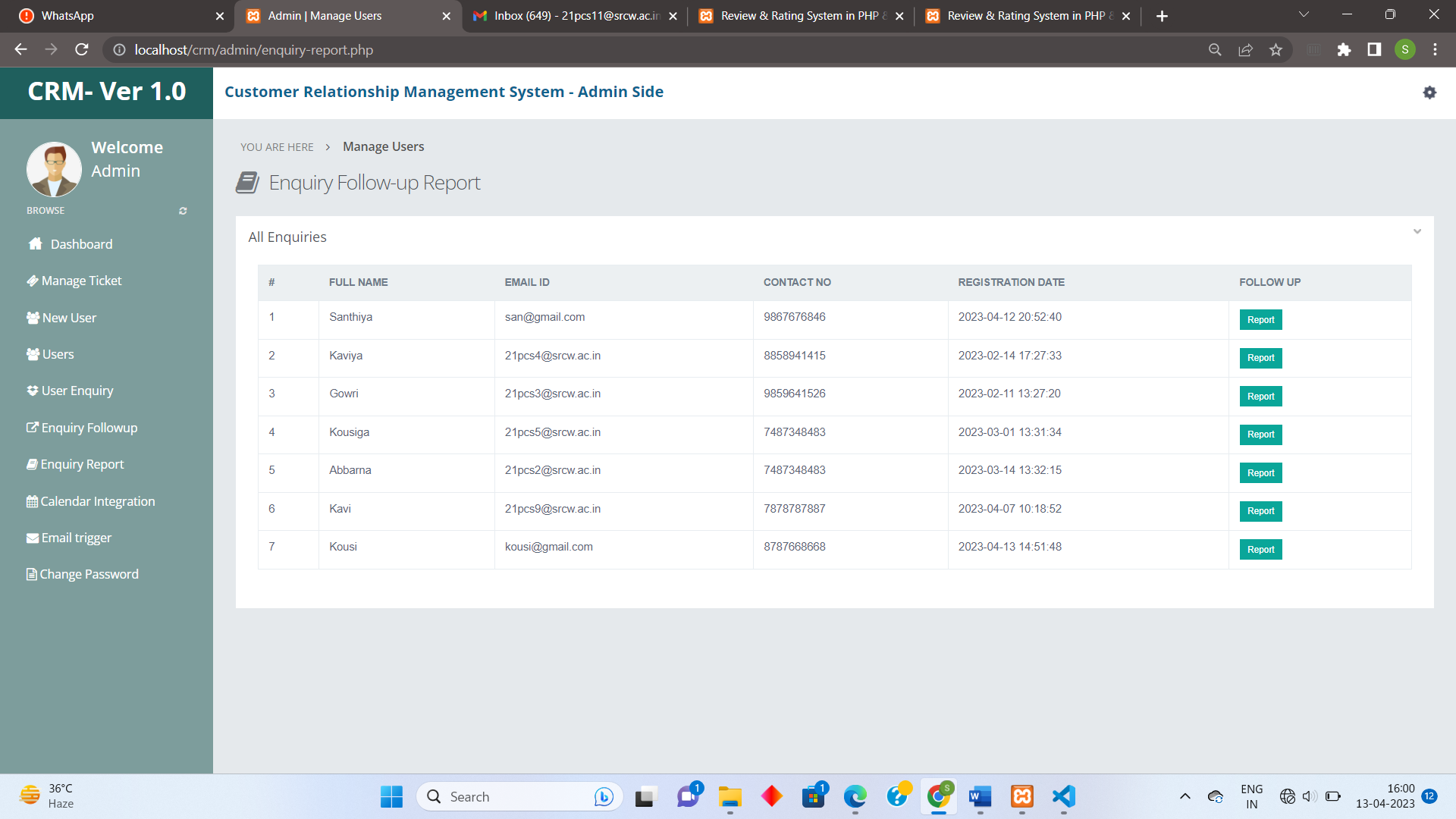
Customer enquiry:



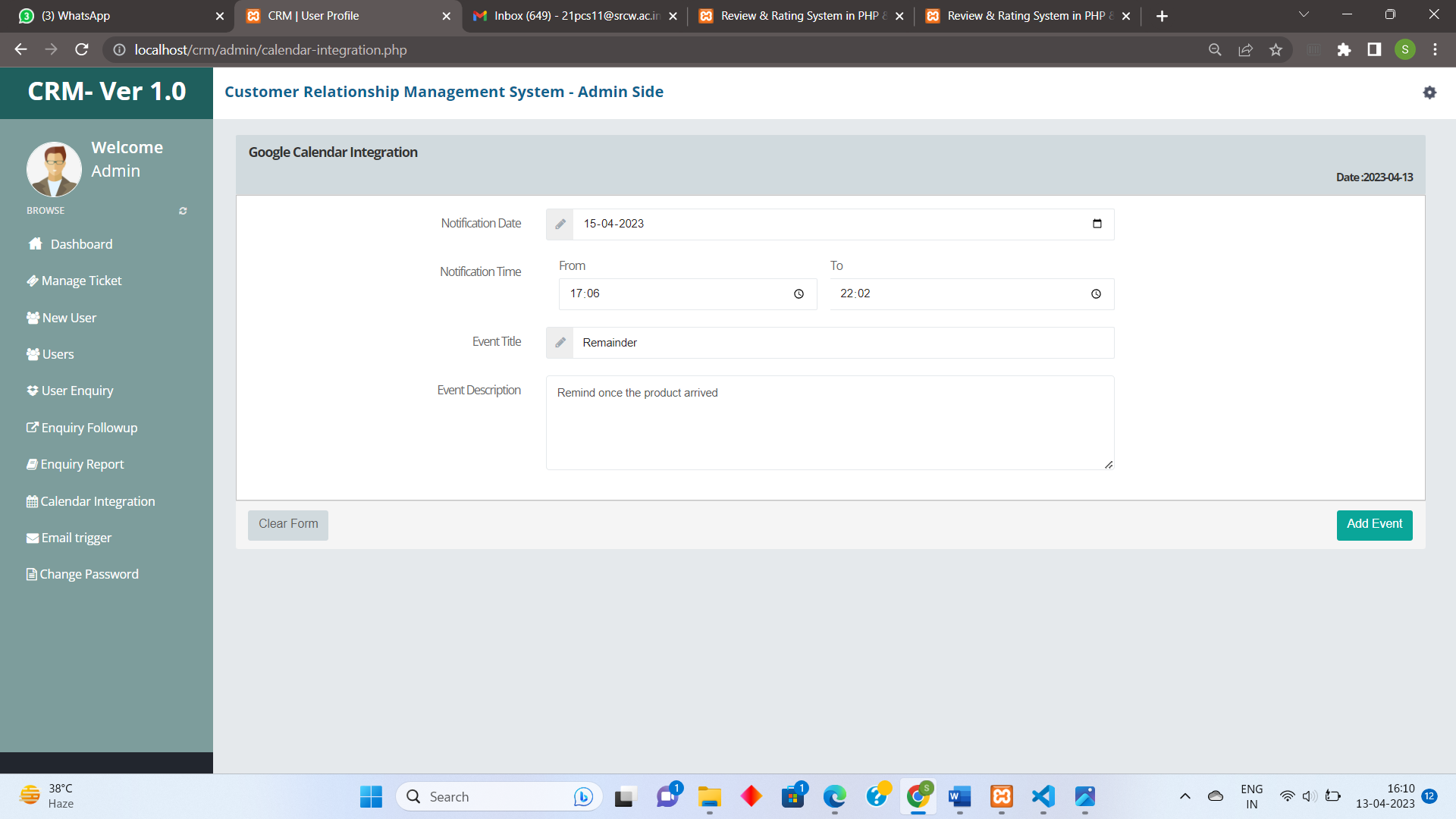
Enquiry follow up:

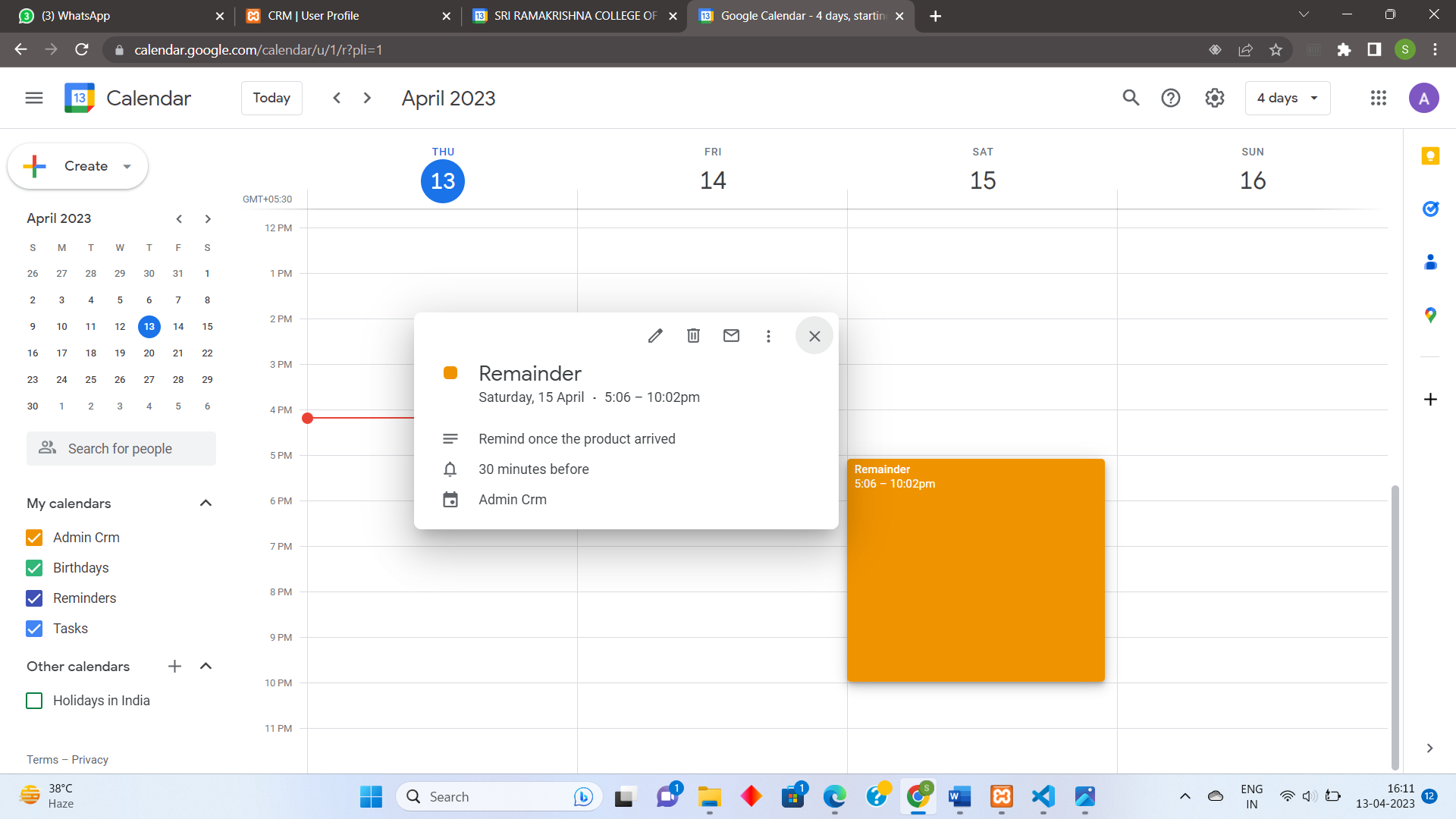


Enquiry report:

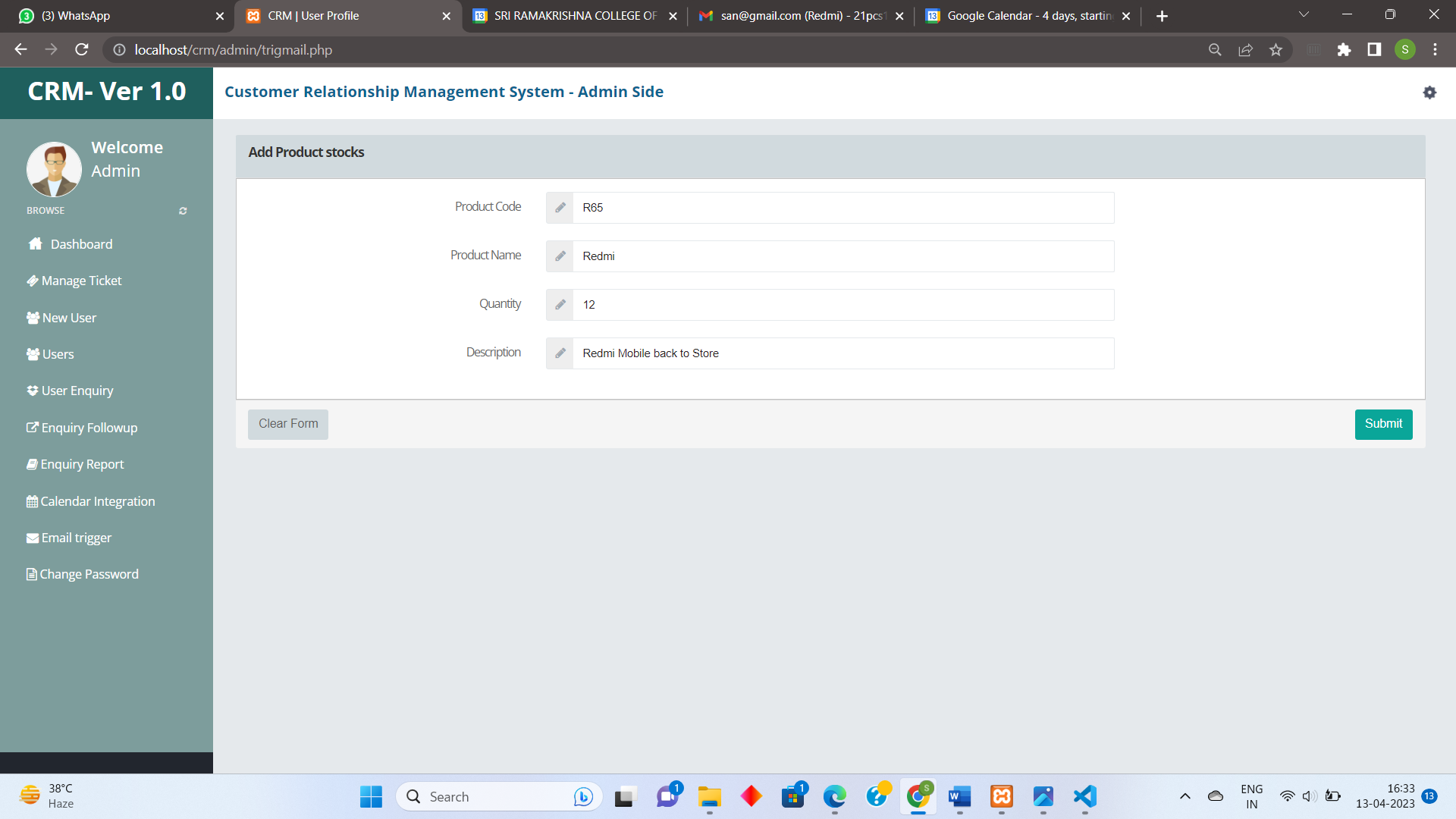


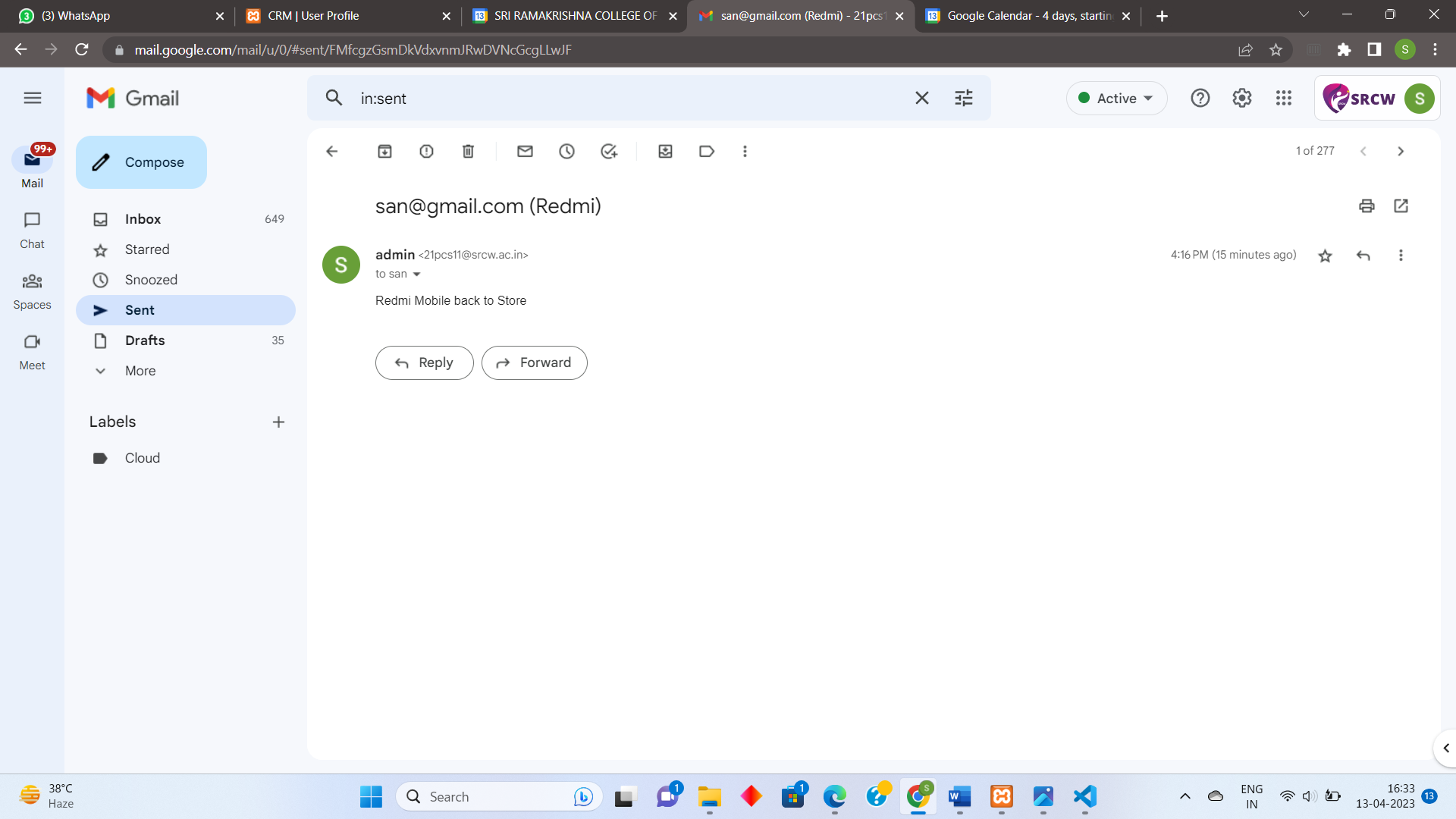
Calendar Integration



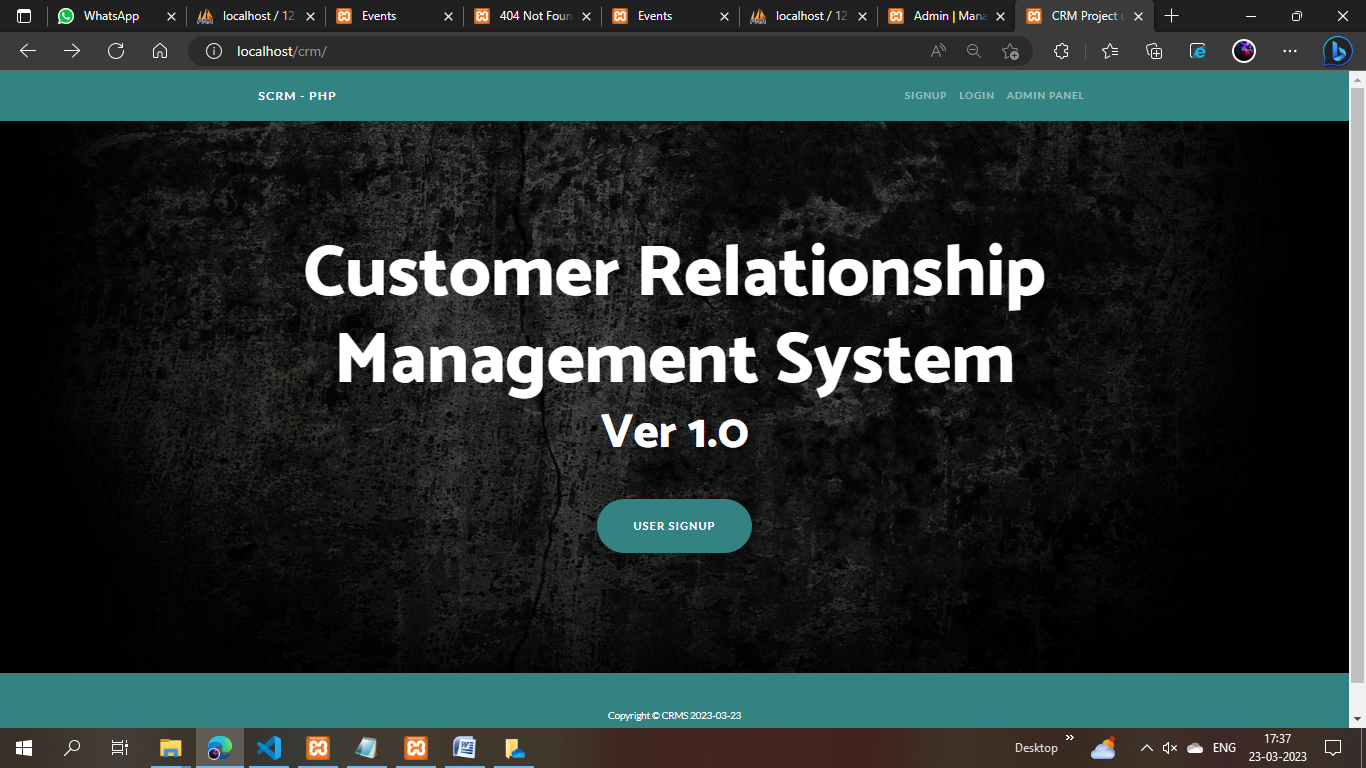


Email Trigger

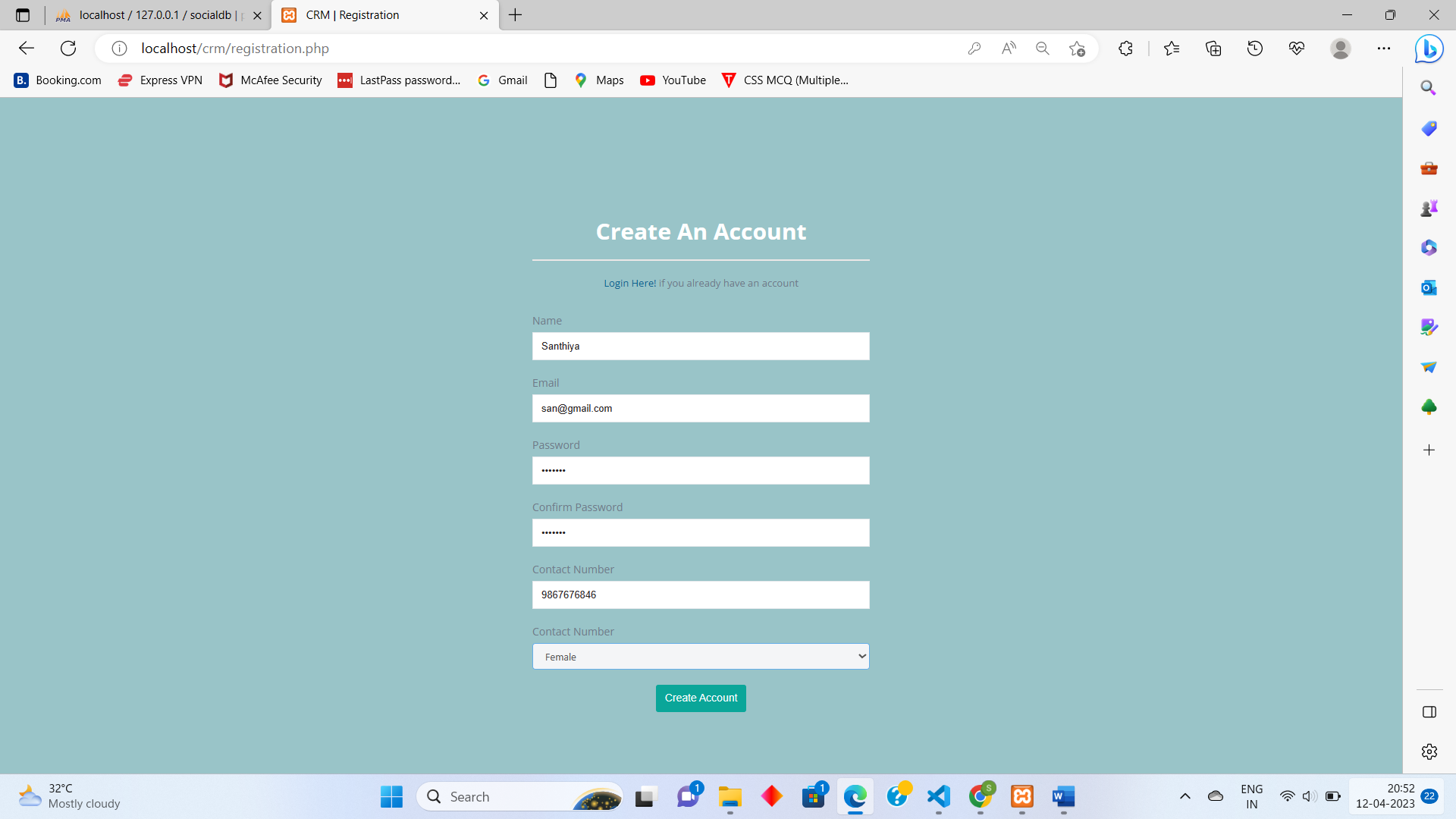


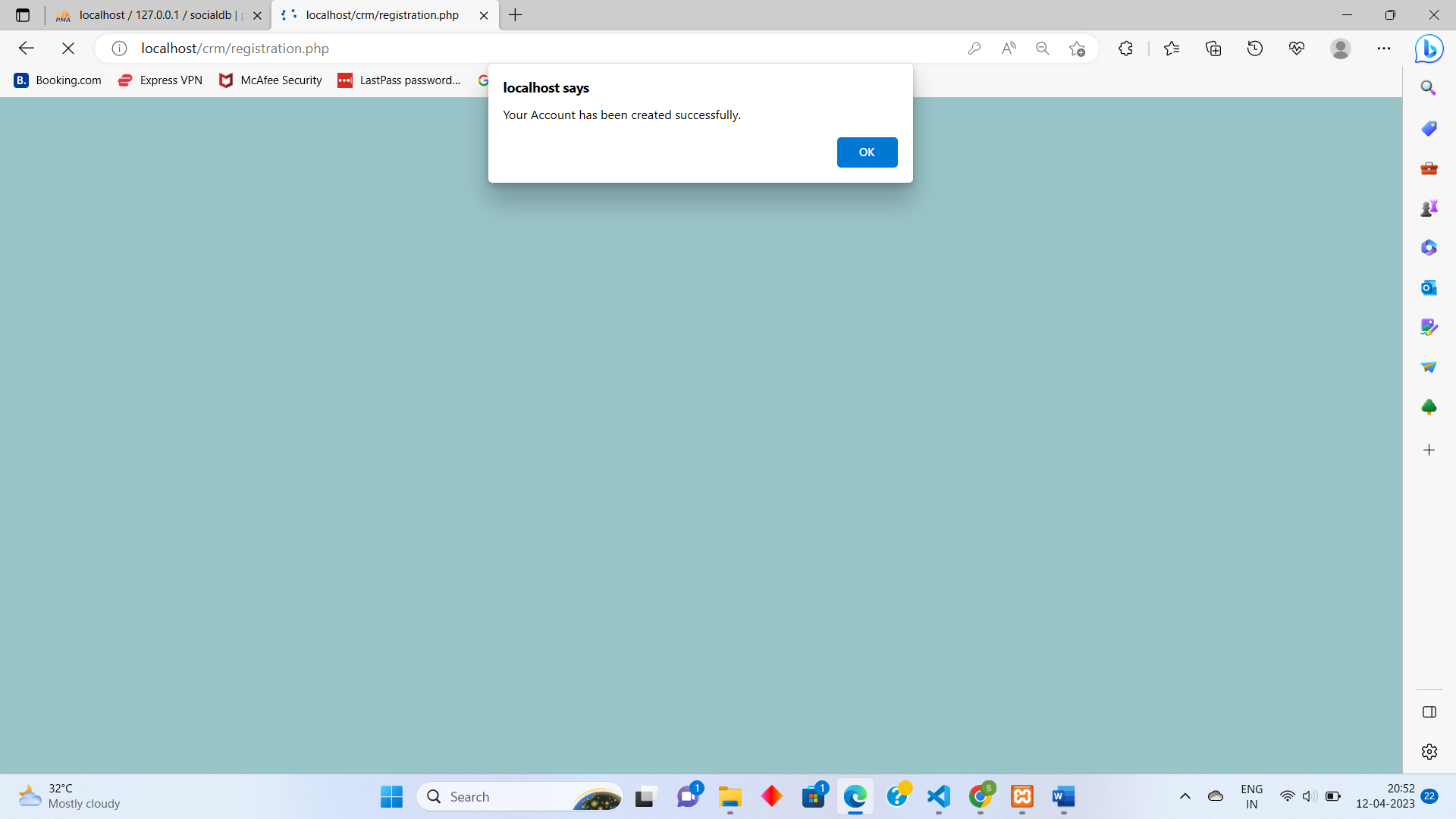


User side:

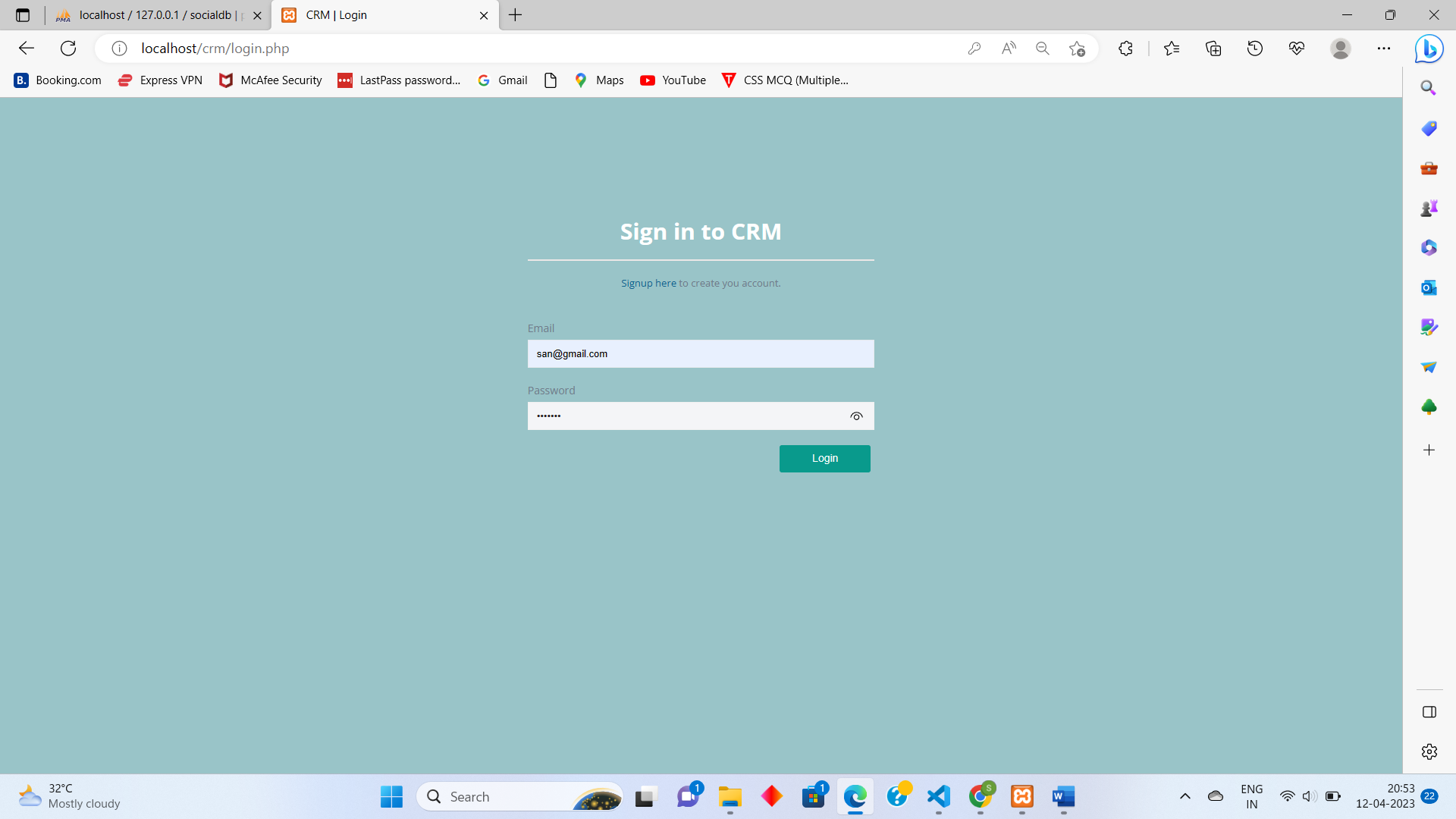


User sign up:

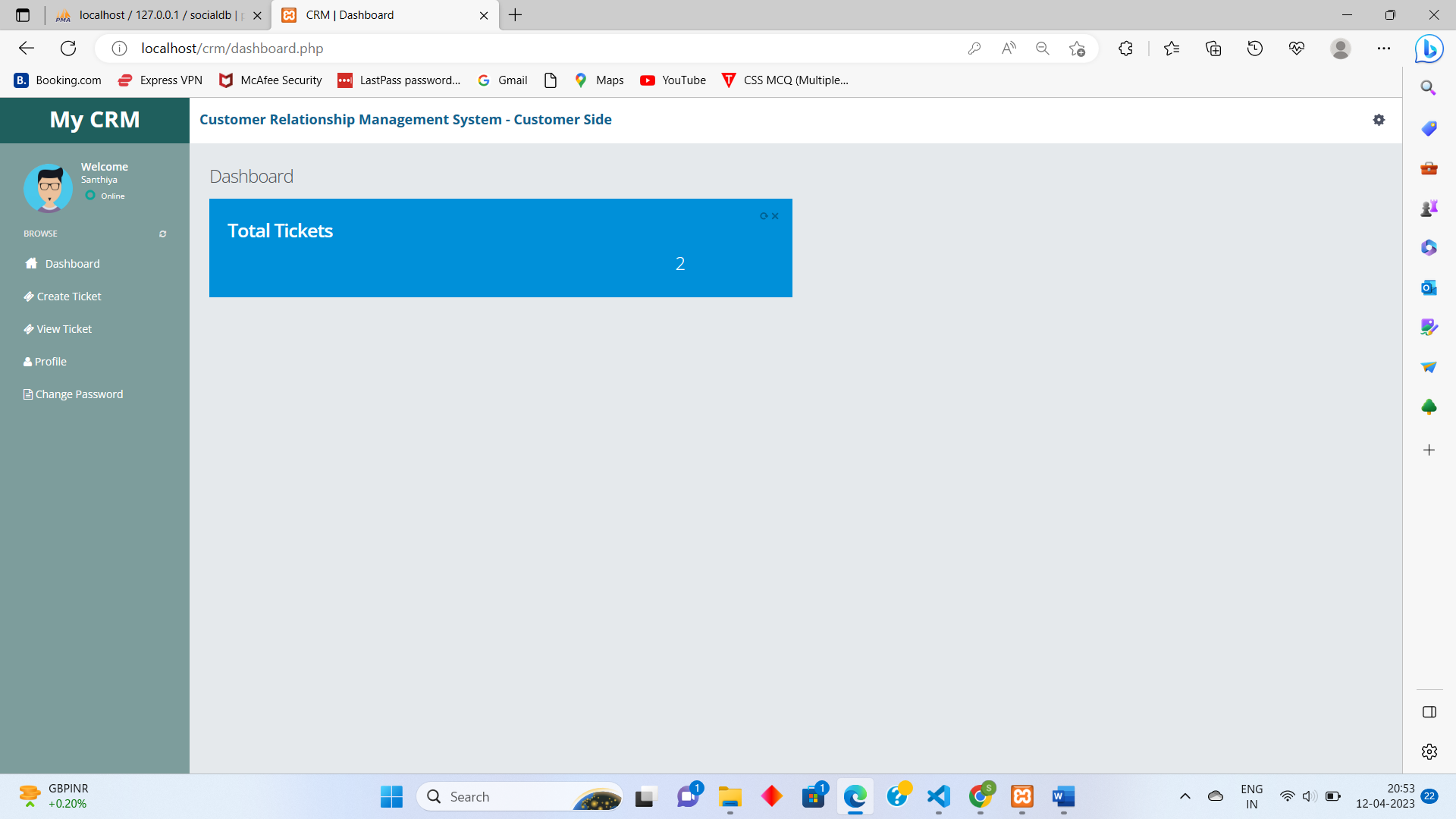




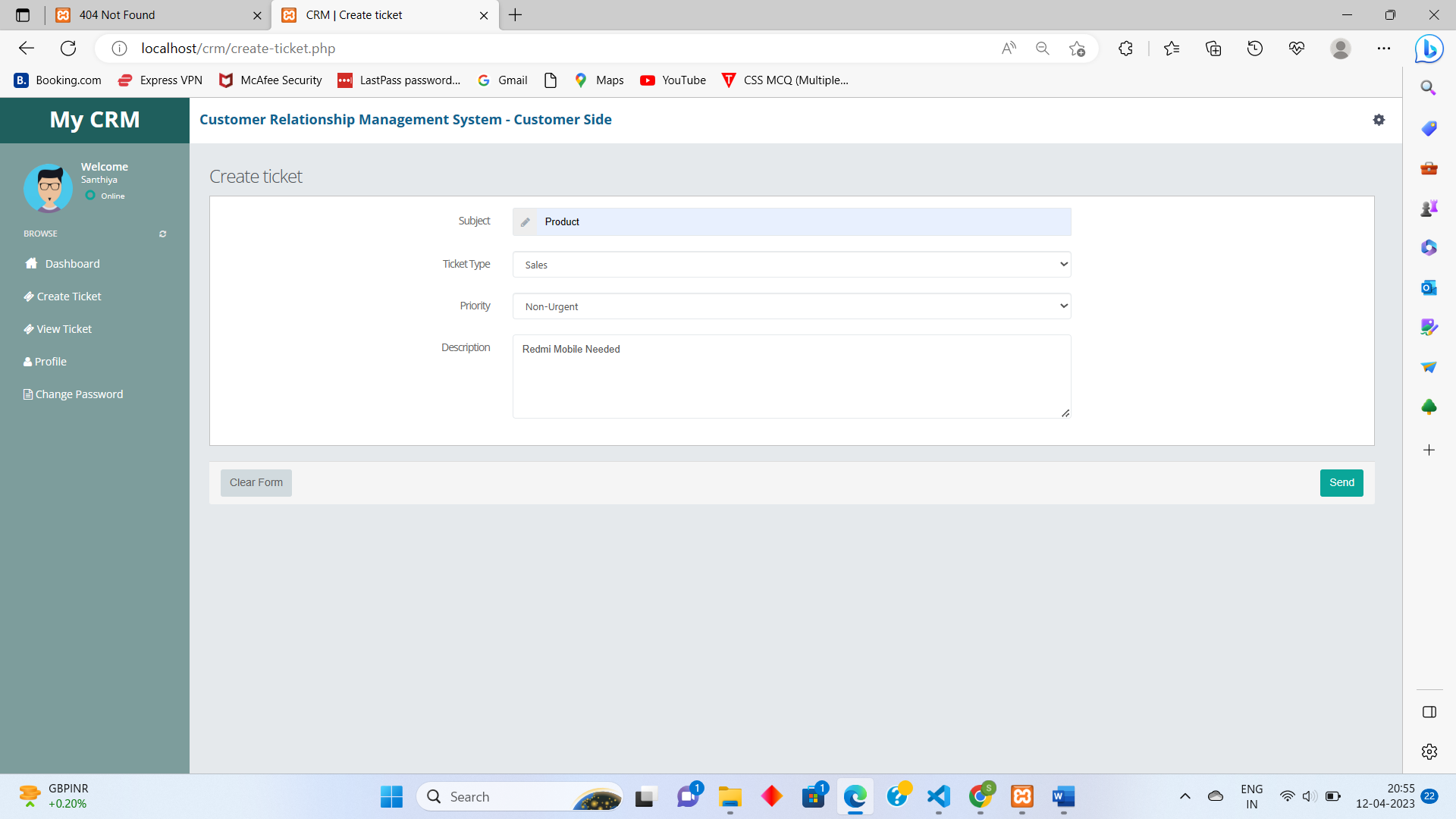
User login:

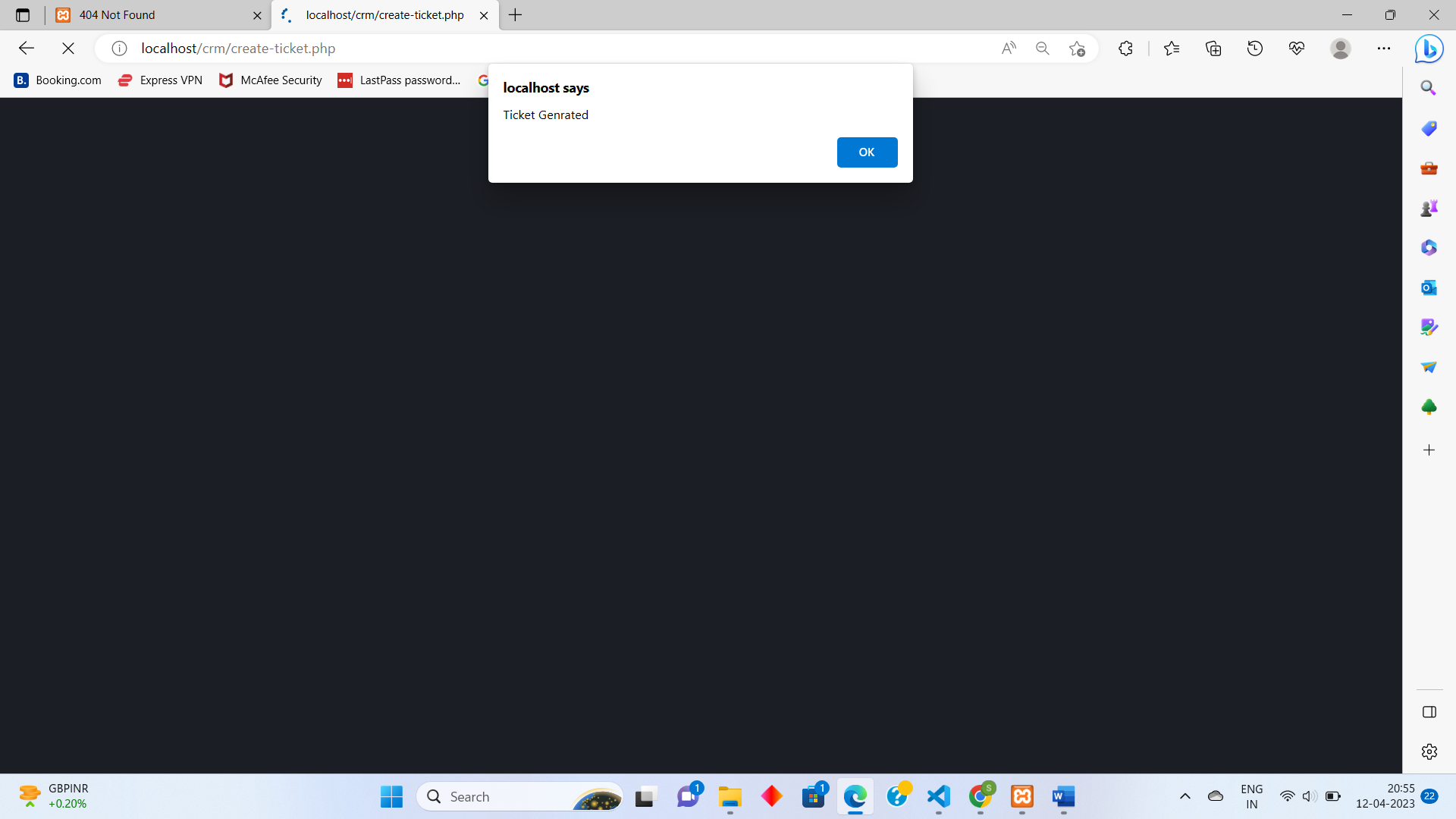


User dashboard:

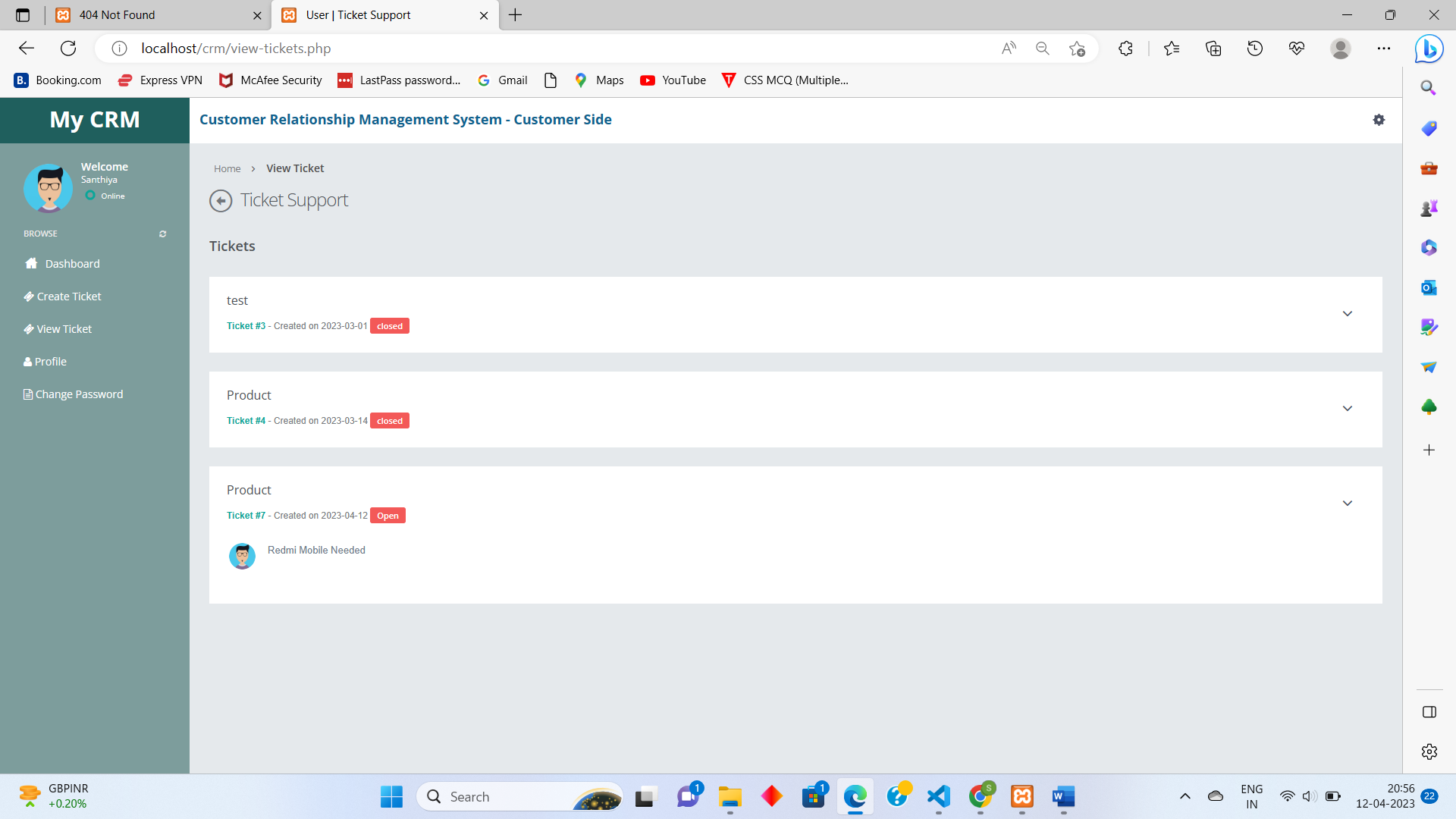


Create ticket:

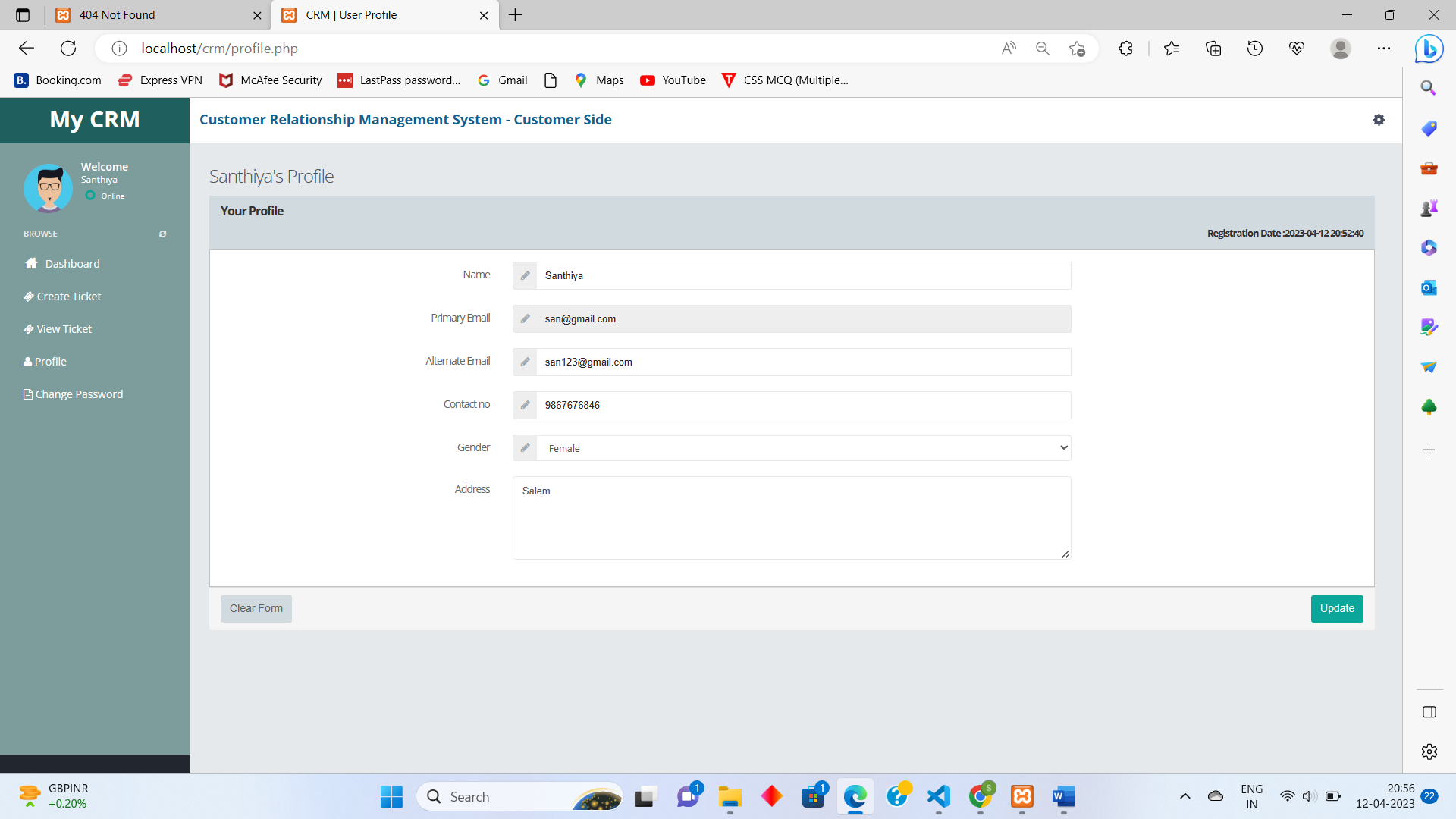


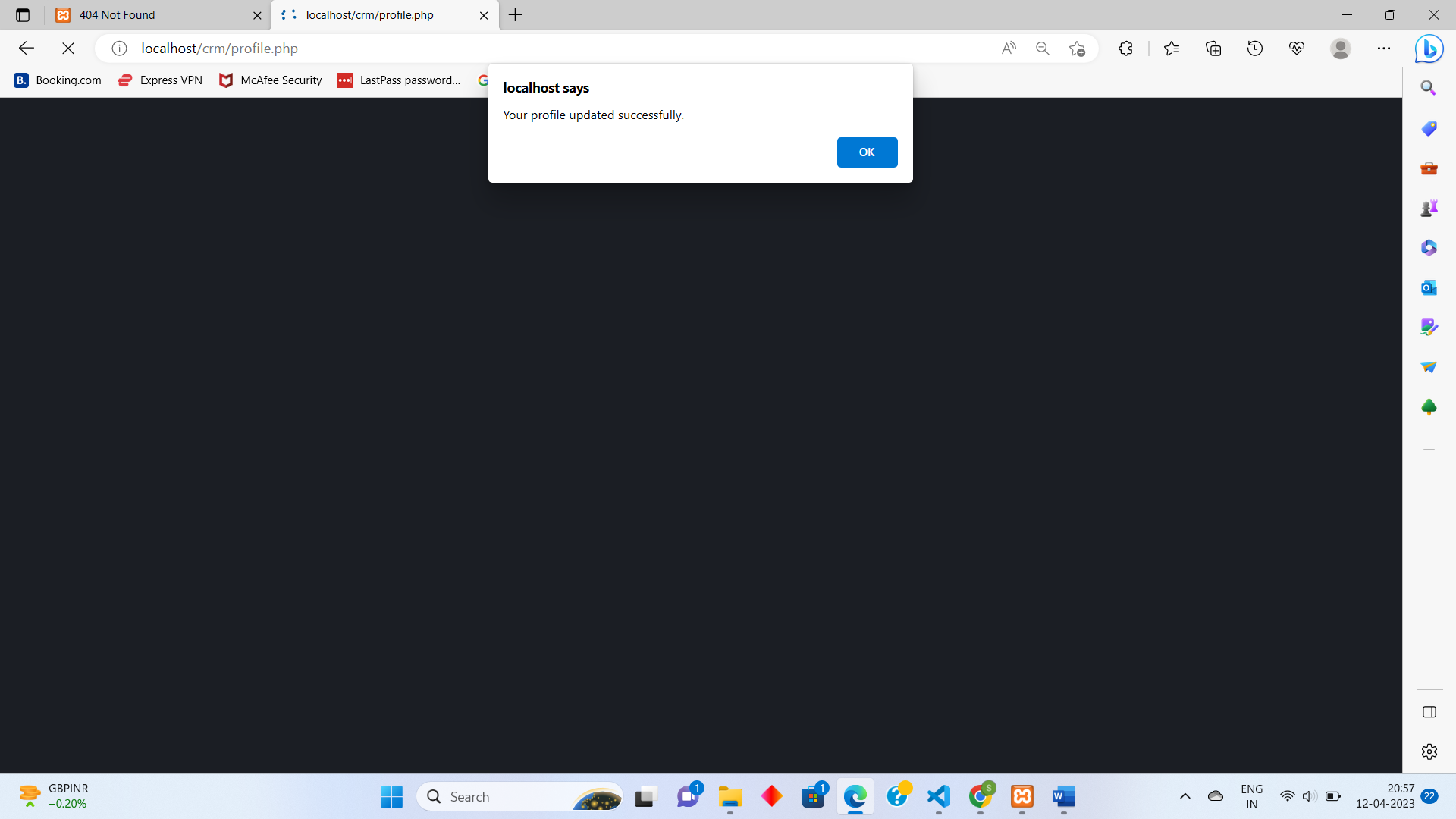


View ticket:



User profile:





SAMPLE SOURCE CODE:

User Registration

<?php

session\_start();

error\_reporting(0);

include("dbconnection.php");

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$name = $\_POST['name'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$mobile = $\_POST['phone'];

$gender = $\_POST['gender'];

$query = mysqli\_query($con, "select email from user where email='$email'");

$num = mysqli\_fetch\_array($query);

if ($num > 1) {

echo "<script>alert('Email already register with us. Please try with diffrent email id.');</script>";

echo "<script>window.location.href='registration.php'</script>";

} else {

mysqli\_query($con, "insert into user(name,email,password,mobile,gender) values('$name','$email','$password','$mobile','$gender')");

echo "<script>alert('Your Account has been created successfully.');</script>";

echo "<script>window.location.href='login.php'</script>";

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html;charset=UTF-8" />

<meta charset="utf-8" />

<title>CRM | Registration</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />

<meta content="" name="description" />

<meta content="" name="author" />

<link href="assets/plugins/pace/pace-theme-flash.css" rel="stylesheet" type="text/css" media="screen" />

<link href="assets/plugins/boostrapv3/css/bootstrap.min.css" rel="stylesheet" type="text/css" />

<link href="assets/plugins/boostrapv3/css/bootstrap-theme.min.css" rel="stylesheet" type="text/css" />

<link href="assets/plugins/font-awesome/css/font-awesome.css" rel="stylesheet" type="text/css" />

<link href="assets/css/animate.min.css" rel="stylesheet" type="text/css" />

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

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

<link href="assets/css/custom-icon-set.css" rel="stylesheet" type="text/css" />

<script type="text/javascript">

function checkpass() {

if (document.signup.password.value != document.signup.cpassword.value) {

alert('New Password and Re-Password field does not match');

document.signup.cpassword.focus();

return false;

}

return true;

}

</script>

</head>

<body class="error-body no-top">

<div class="container">

<div class="login-container">

<div class="col-md-5">

<h2 class="text-center text-white"><strong>Create An Account</strong></h2>

<hr style="border-color:#ebe7e7">

<p class="text-center"><a href="login.php">Login Here!</a> if you already have an account</p>

</div>

<div class="col-md-5 "> <br>

<form id="signup" name="signup" class="login-form" onsubmit="return checkpass();" method="post">

<div class="form-group">

<label for="name" class="control-label">Name</label>

<input type="text" class="form-control rounded-0" id="name" name="name" required="required">

</div>

<div class="form-group">

<label for="email" class="control-label">Email</label>

<input type="text" class="form-control rounded-0" id="email" name="email" required="required">

</div>

<div class="form-group">

<label for="password" class="control-label">Password</label>

<input type="password" class="form-control rounded-0" id="password" name="password" required="required">

</div>

<div class="form-group">

<label for="password" class="control-label">Confirm Password</label>

<input type="password" class="form-control rounded-0" id="cpassword" name="cpassword" required="required">

</div>

<div class="form-group">

<label for="phone" class="control-label">Contact Number</label>

<input type="text" pattern="[0-9]{10}" class="form-control rounded-0" id="phone" name="phone" required="required">

</div>

<div class="form-group">

<label for="gender" class="control-label">Contact Number</label>

<select class="form-control" name="gender" id="gender" required>

<option value="male">Male</option>

<option value="female">Female</option>

</select>

</div>

<div class="form-group text-center">

<button class="btn btn-primary rounded-pill">Create Account</button>

</div>

</form>

</div>

</div>

</div>

<script src="assets/plugins/jquery-1.8.3.min.js" type="text/javascript"></script>

<script src="assets/plugins/bootstrap/js/bootstrap.min.js" type="text/javascript"></script>

<script src="assets/plugins/pace/pace.min.js" type="text/javascript"></script>

<script src="assets/plugins/jquery-validation/js/jquery.validate.min.js" type="text/javascript"></script>

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

</body>

</html>

**CREATE NEW TICKET:**

<?php

session\_start();

//echo $\_SESSION['id'];

//$\_SESSION['msg'];

include("dbconnection.php");

include("checklogin.php");

check\_login();

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

$count\_my\_page = ("hitcounter.txt");

$hits = file($count\_my\_page);

$hits[0]++;

$fp = fopen($count\_my\_page, "w");

fputs($fp, "$hits[0]");

fclose($fp);

$tid = $hits[0];

$email = $\_SESSION['login'];

$subject = $\_POST['subject'];

$tt = $\_POST['tasktype'];

$priority = $\_POST['priority'];

$ticket = $\_POST['description'];

//$ticfile=$\_FILES["tfile'"]["name"];

$st = "Open";

$pdate = date('Y-m-d');

//move\_uploaded\_file($\_FILES["tfile"]["tmp\_name"],"ticketfiles/".$\_FILES["tfile"]["name"]);

$a = mysqli\_query($con, "insert into ticket(ticket\_id,email\_id,subject,task\_type,prioprity,ticket,status,posting\_date) values('$tid','$email','$subject','$tt','$priority','$ticket','$st','$pdate')");

if ($a) {

echo "<script>alert('Ticket Genrated'); location.replace(document.referrer)</script>";

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html;charset=UTF-8" />

<meta charset="utf-8" />

<title>CRM | Create ticket</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />

<meta content="" name="description" />

<meta content="" name="author" />

<link href="assets/plugins/pace/pace-theme-flash.css" rel="stylesheet" type="text/css" media="screen" />

<link href="assets/plugins/boostrapv3/css/bootstrap.min.css" rel="stylesheet" type="text/css" />

<link href="assets/plugins/boostrapv3/css/bootstrap-theme.min.css" rel="stylesheet" type="text/css" />

<link href="assets/plugins/font-awesome/css/font-awesome.css" rel="stylesheet" type="text/css" />

<link href="assets/css/animate.min.css" rel="stylesheet" type="text/css" />

<link href="assets/plugins/jquery-scrollbar/jquery.scrollbar.css" rel="stylesheet" type="text/css" />

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

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

<link href="assets/css/custom-icon-set.css" rel="stylesheet" type="text/css" />

</head>

<body class="">

<?php include("header.php"); ?>

<div class="page-container row-fluid">

<?php include("leftbar.php"); ?>

<div class="clearfix"></div>

<!-- END SIDEBAR MENU -->

</div>

</div>

<!-- END SIDEBAR -->

<!-- BEGIN PAGE CONTAINER-->

<div class="page-content">

<!-- BEGIN SAMPLE PORTLET CONFIGURATION MODAL FORM-->

<div id="portlet-config" class="modal hide">

<div class="modal-header">

<button data-dismiss="modal" class="close" type="button"></button>

<h3>Widget Settings</h3>

</div>

<div class="modal-body"> Widget settings form goes here </div>

</div>

<div class="clearfix"></div>

<div class="content">

<div class="page-title">

<h3>Create ticket</h3>

<div class="row">

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

<form class="form-horizontal" name="form1" method="post" action="" onSubmit="return valid();">

<div class="panel panel-default">

<div class="panel-body bg-white">

<?php if (isset($\_SESSION['msg1'])) : ?>

<p align="center" style="color:#FF0000"><?= $\_SESSION['msg1']; ?><?= $\_SESSION['msg1'] = ""; ?></p>

<?php endif; ?>

<div class="form-group">

<label class="col-md-3 col-xs-12 control-label">Subject</label>

<div class="col-md-6 col-xs-12">

<div class="input-group">

<span class="input-group-addon"><span class="fa fa-pencil"></span></span>

<input type="text" name="subject" id="subject" value="" required class="form-control" />

</div>

</div>

</div>

<div class="form-group">

<label class="col-md-3 col-xs-12 control-label">Ticket Type</label>

<div class="col-md-6 col-xs-12">

<select name="tasktype" class="form-control select" required>

<option value="#">--Select--</option>

<option value="Sales">Sales</option>

<option value="Service">Service</option>

<option value="General">General</option>

</select>

</div>

</div>

<div class="form-group">

<label class="col-md-3 col-xs-12 control-label">Priority</label>

<div class="col-md-6 col-xs-12">

<select name="priority" class="form-control select">

<option value="">Choose your Priority</option>

<option value="important">Important</option>

<option value="urgent(functional problem)">Urgent (Functional Problem)</option>

<option value="non-urgent">Non-Urgent</option>

<option value="question">Question</option>

</select>

</div>

</div>

<div class="form-group">

<label class="col-md-3 col-xs-12 control-label">Description</label>

<div class="col-md-6 col-xs-12">

<textarea name="description" required class="form-control" rows="5"></textarea>

</div>

</div>

</div>

</div>

<div class="panel-footer">

<button class="btn btn-default">Clear Form</button>

<input type="submit" value="Send" name="send" class="btn btn-primary pull-right">

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

<script src="assets/plugins/jquery-1.8.3.min.js" type="text/javascript"></script>

<script src="assets/plugins/jquery-ui/jquery-ui-1.10.1.custom.min.js" type="text/javascript"></script>

<script src="assets/plugins/bootstrap/js/bootstrap.min.js" type="text/javascript"></script>

<script src="assets/plugins/breakpoints.js" type="text/javascript"></script>

<script src="assets/plugins/jquery-unveil/jquery.unveil.min.js" type="text/javascript"></script>

<script src="assets/plugins/jquery-block-ui/jqueryblockui.js" type="text/javascript"></script>

<script src="assets/plugins/jquery-scrollbar/jquery.scrollbar.min.js" type="text/javascript"></script>

<script src="assets/plugins/pace/pace.min.js" type="text/javascript"></script>

<script src="assets/plugins/jquery-numberAnimate/jquery.animateNumbers.js" type="text/javascript"></script>

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

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

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

</body>

</html>