**CHAPTER 1**

**INTRODUCTION TO DATABASE MANAGEMENT SYSTEM**

**1.1: INTRODUCTION**

Formally, a "database" refers to a set of related data and the way it is organized. Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

Because of the close relationship between them, the term "database" is often used casually to refer to both a database and the DBMS used to manipulate it.

Outside the world of professional [information technology](https://en.wikipedia.org/wiki/Information_technology), the term database is often used to refer to any collection of related data (such as a [spreadsheet](https://en.wikipedia.org/wiki/Spreadsheet) or a card index) as however size and usage requirements typically necessitate use of a database management system.

Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

* **Data definition** – Creation, modification and removal of definitions that define the organization of the data.
* **Update** – Insertion, modification, and deletion of the actual data.
* **Retrieval** – Providing information in a form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
* **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.

Both a database and its DBMS conform to the principles of a particular [database model](https://en.wikipedia.org/wiki/Database_model). "Database system" refers collectively to the database model, database management system, and database.

Physically, database [servers](https://en.wikipedia.org/wiki/Server_(computing)) are dedicated computers that hold the actual databases and run only the DBMS and related software. Database servers are usually [multiprocessor](https://en.wikipedia.org/wiki/Multiprocessor) computers, with generous memory and [RAID](https://en.wikipedia.org/wiki/Redundant_array_of_independent_disks) disk arrays used for stable storage. RAID is used for recovery of data if any of the disks fail. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most [database applications](https://en.wikipedia.org/wiki/Database_application). DBMSs may be built around a custom [multitasking](https://en.wikipedia.org/wiki/Computer_multitasking) [kernel](https://en.wikipedia.org/wiki/Kernel_(computing)) with built-in [networking](https://en.wikipedia.org/wiki/Computer_network) support, but modern DBMSs typically rely on a standard [operating system](https://en.wikipedia.org/wiki/Operating_system) to provide these functions. Since DBMSs comprise a significant [market](https://en.wikipedia.org/wiki/Market_(economics)), computer and storage vendors often take into account DBMS requirements in their own development plans.

Databases and DBMSs can be categorized according to the database model(s) that they support (such as relational or XML), the type(s) of computer they run on (from a server cluster to a mobile phone), the [query language](https://en.wikipedia.org/wiki/Query_language)(s) used to access the database (such as SQL or [XQuery](https://en.wikipedia.org/wiki/XQuery)), and their internal engineering, which affects performance, [scalability](https://en.wikipedia.org/wiki/Scalability), resilience, and security.

**1.2: HISTORY**

The sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. These performance increases were enabled by the technology progress in the areas of [processors](https://en.wikipedia.org/wiki/Central_processing_unit), [computer memory](https://en.wikipedia.org/wiki/Computer_memory), [computer storage](https://en.wikipedia.org/wiki/Computer_data_storage), and [computer networks](https://en.wikipedia.org/wiki/Computer_network). The development of database technology can be divided into three eras based on data model or structure: [navigational](https://en.wikipedia.org/wiki/Navigational_database), SQL/[relational](https://en.wikipedia.org/wiki/Relational_database), and post-relational.

The two main early navigational data models were the [hierarchical model](https://en.wikipedia.org/wiki/Hierarchical_database_model) and the [CODASYL](https://en.wikipedia.org/wiki/CODASYL) model ([network model](https://en.wikipedia.org/wiki/Network_model)).

The [relational model](https://en.wikipedia.org/wiki/Relational_model), first proposed in 1970 by [Edgar F. Codd](https://en.wikipedia.org/wiki/Edgar_F._Codd), departed from this tradition by insisting that applications should search for data by content, rather than by following links. The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale [data processing](https://en.wikipedia.org/wiki/Data_processing) applications, and as of 2018 they remain dominant: [IBM DB2](https://en.wikipedia.org/wiki/IBM_DB2), [Oracle](https://en.wikipedia.org/wiki/Oracle_database), [MySQL](https://en.wikipedia.org/wiki/MySQL), and [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server) are the most searched [DBMS](https://en.wikipedia.org/wiki/DBMS). The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models. [Object databases](https://en.wikipedia.org/wiki/Object_database) were developed in the 1980s to overcome the inconvenience of [object-relational impedance mismatch](https://en.wikipedia.org/wiki/Object-relational_impedance_mismatch), which led to the coining of the term "post-relational" and also the development of hybrid [object-relational databases](https://en.wikipedia.org/wiki/Object-relational_database).

The next generation of post-relational databases in the late 2000s became known as [NoSQL](https://en.wikipedia.org/wiki/NoSQL) databases, introducing fast [key-value stores](https://en.wikipedia.org/wiki/Key-value_store) and [document-oriented databases](https://en.wikipedia.org/wiki/Document-oriented_database). A competing "next generation" known as [NewSQL](https://en.wikipedia.org/wiki/NewSQL) databases attempted new implementations that retained the relational/SQL model while aiming to match the high performance of NoSQL compared to commercially available relational DBMSs.

**1.3: CHARACTERISTICS OF DATABASE APPROACH**

* Represent Some Aspects of real world applications
* Manages Information
* Easy Operation implementation
* Multiple views of database
* Data for specific purpose
* It has Users of Specific interest
* Self-Describing nature
* Logical relationship between records and data

**1.4: APPLICATION OF DBMS**

A Database management system is a computerized record-keeping system. It is a repository or a container for collection of computerized data files. The overall purpose of DBMS is to allow the users to define, store, retrieve and update the information contained in the database on demand. Information can be anything that is of significance to an individual or organization.

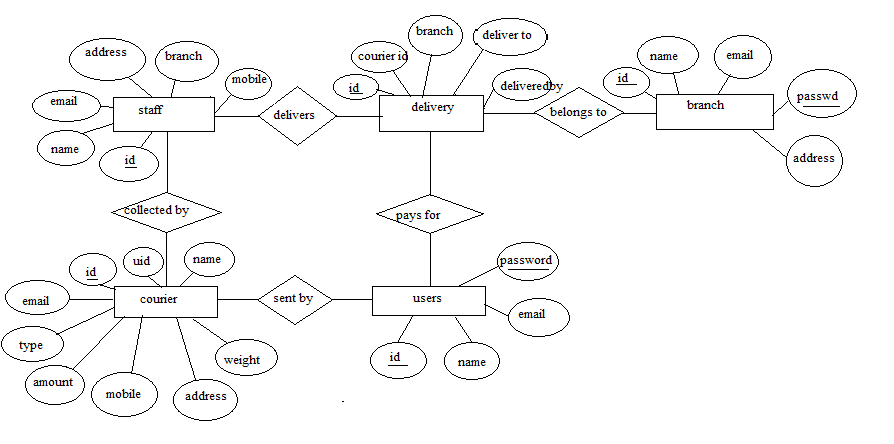
**Databases touch all aspects of our lives. Some of the major areas of application are as follows:**

* Banking
* Airlines
* Universities
* Manufacturing And Selling
* Human Resources

**CHAPTER 2**

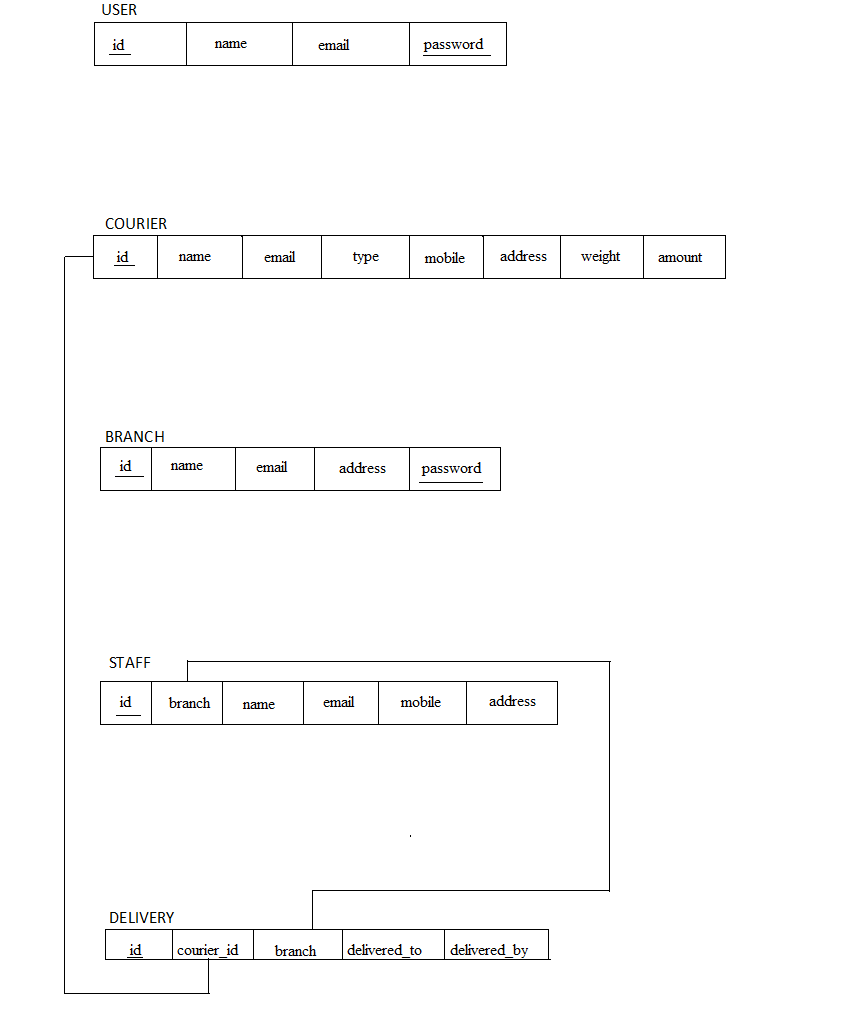
**SCHEMA AND ER DIAGRAM**

**2.1 ER Diagram:**

****

**Fig 2.1:** ER Diagram

**2.2 Schema Diagram:**

****

**Fig 2.2:** Schema diagram

**CHAPTER 3**

**RESOURCE REQUIREMENTS**

**3.1 HARDWARE REQUIREMENTS**

Processor : Intel CORE i3

RAM : 4 GB

Hard disk : 238 GB

Monitor     : VGA compatible (CRT or LCD-TFT)

**3.2 SOFTWARE REQUIREMENTS**

* **XAMPP server**
* **Front End :** HTML 5 ,PHP
* **Back End :** MYSQL

**CHAPTER4**

**DESCRIPTION OF TOOLS AND TECHNOLOGIES**

**4.1 DESCRIPTION OF TOOLS**

* + 1. **XAMPP Server**

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). Since XAMPP is simple, lightweight. Apache distribution it is extremely easy for developers to create a local web server for testing and deployment purposes. Everything you needed is to set up a web server-server application (Apache), database (MariaDB), and scripting language (PHP). XAMPP works equally well on Linux, Mac, and Windows.

**4.1.2** **MySQL SERVER**

It is an open-source relational database management system (RDBMS). Its name is a combination of “My”, the name of co-founder Michael Widenius’s daughter, and “SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

The MySQL server package will install the MySQL database server which can interact with using a MySQL client. User can use the MySQL client to send commands to any MySQL server; on a remote computer The MySQL server is used to persist the data and provide a query interface for it (SQL). The MySQL client’s purpose is to allow you to use that query interface. The client package also comes with utilities that allow you to easily backup/restore data and administer the server.

**4.2 DESCRIPTION OF TECHNOLOGIES**

* + 1. **HTML**

HTML stands for Hypertext Markup Language. It is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages, with HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page.

It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Browsers do not display the HTML tags, but use them to interpret the content of the page.

* + 1. **Cascading Style Sheets (CSS)**

It is a style sheet language used for describing the presentation of a document written in a markup language Although most often used to set the visual style of web pages and user interfaces written in HTML, the language can be applied to any document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content makes it possible to present the same markup page in different styles.

* + 1. **PHP**

It is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. PHP was originally created by Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Development Team. PHP stands for the acronym: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the ZendEngine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

**CHAPTER 5**

**IMPLEMENTATION AND CODE**

**5.1 INDEX.HTML**

<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

<?php

session\_start();

if(isset($\_SESSION["bid"])){

header('Location:BranchPage.php');

}

if(isset($\_SESSION["uid"])){

header('Location:UserPage.php');

}

?>

<html>

<head>

<meta charset="UTF-8">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1"><link rel="icon" href="images/homepage/favicon.ico" type="image/x-icon">

<title> CourierManagement</title>

<meta name="author" content="Audenberg Technologies (www.audenberg.com)" />

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

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

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

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css" integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU" crossorigin="anonymous">

<link rel="stylesheet" href="css/bootstrap.min.css">

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

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

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

<link href="https://fonts.googleapis.com/css?family=Kodchasan" rel="stylesheet">

</head>

<body>

<?php

// put your code here

include 'navBar.php';

include 'signinModals.php';

?>

<div class="" id="myEmployerModal" tabindex="-1" role="dialog" aria-labelledby="myEmployerModalLabel" style="background-image:url('images/bg.jpg'); background-size: cover; background-repeat: no-repeat; height: 100vh;">

<div class="modal-dialog" role="document" >

<div class="modal-content" style=" background-color:rgba(225,225,225,0.5) ; box-shadow: 0px 0px 20px #1e1e1e; padding: 12%;">

<div class="modal-header">

<button id="empSignInCloseBtn" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h4 class="modal-title" id="myEmployerModalLabel">SignIn</h4>

</div>

<div class="modal-body">

<div id="cd-login"> <!-- log in form -->

<form class="cd-form" method="post" action="signin.php">

<p class="fieldset">

<label class="image-replace cd-email" for="email">E-mail</label>

<input class="full-width has-padding has-border" id="email" name="email" type="email" placeholder="E-mail">

<span class="cd-error-message">Error message here!</span>

</p>

<p class="fieldset">

<label class="image-replace cd-password" for="epass">Password</label>

<input class="full-width has-padding has-border" id="password" name="password" type="password" placeholder="Password">

<a href="#0" class="hide-password">Show</a>

<div id="loginresult" style="display:none;">Error message here!</div>

</p>

<input type="hidden" id="currentPage" name="currentPage" value="<?php echo $\_SERVER['PHP\_SELF']; ?>"

<p class="fieldset">

<input id="loginsubmit" class="full-width" type="submit" name="loginsubmit" id="login" value="Login">

</p>

</form>

<p class="cd-form-bottom-message">

<button id="regNowBtn" class="btn btn-default" data-toggle="modal" data-target="#empsignup" style="color: brown;" >

Register Now</button></p>

<button id="regEmpModalBtn" style="display:none;" data-toggle="modal" data-target="#empsignup" >

</button>

<!-- <a href="#0" class="cd-close-form">Close</a> -->

</div>

</div>

<div class="modal-footer">

</div>

</div>

</div>

</div>

</body>

</html>

**Add courier:**

<?php

include 'connect.php';

session\_start();

if(!isset($\_SESSION["uid"])){

header('Location:index.php');

}

$name =$email= $mobile=$address=$courierW=$courierA=$courierT="";

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

if(isset($\_POST["cname"])){

$name = $\_POST["cname"];

$email = $\_POST["cemail"];

$mobile = $\_POST["cMob"];

$address = $\_POST["cAdd"];

$courierW = $\_POST["cWeight"];

$courierA = $\_POST["cAmount"];

$courierT = $\_POST["cType"];

$userId = $\_SESSION['uid'];

$sql = "INSERT INTO courier (name,email,mobile,address,weight,amount,type,uid)

VALUES ('$name', '$email', '$mobile','$address', '$courierW', '$courierA', '$courierT','$userId')";

echo '/n'.$sql;

if ($conn->query($sql) === TRUE) {

header("Location:viewCouriers.php");

}

}}

?>

<html>

<head>

<meta charset="UTF-8">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1"><link rel="icon" href="images/homepage/favicon.ico" type="image/x-icon">

<title> CourierManagement</title>

<meta name="author" content="Audenberg Technologies (www.audenberg.com)" />

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

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

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

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css" integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU" crossorigin="anonymous">

<link rel="stylesheet" href="css/bootstrap.min.css">

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

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

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

<link href="https://fonts.googleapis.com/css?family=Kodchasan" rel="stylesheet">

</head>

<body style=" background-image: url(images/bg2.jpg); background-repeat: no-repeat; background-size: cover;">

<?php

// put your code here

include 'navBar.php';

include 'signinModals.php';

?>

<div class="container" style="padding-top: 3%;">

<div class="row justify-content-center" style="background-color: rgba(225,225,225,0.6); padding: 25px;">

<h2>Send Courier to</h2>

<div class="media-container-column col-lg-8" data-form-type="formoid">

<div data-form-alert="" hidden="">

Thanks for filling out the form!

</div>

<form class="mbr-form" action="addCourier.php" method="post" data-form-title="Mobirise Form">

<div class="row row-sm-offset">

<div class="col-md-4 multi-horizontal" data-for="name">

<div class="form-group">

<label class="form-control-label mbr-fonts-style display-7" for="name-form1-4t">Name</label>

<input type="text" class="form-control" name="cname" data-form-field="Name" required="" id="name-form1-4t">

</div>

</div>

<div class="col-md-4 multi-horizontal" data-for="email">

<div class="form-group">

<label class="form-control-label mbr-fonts-style display-7" for="email-form1-4t">Email</label>

<input type="email" class="form-control" name="cemail" required="" >

</div>

</div>

<div class="col-md-4 multi-horizontal" data-for="phone">

<div class="form-group">

<label class="form-control-label mbr-fonts-style display-7" for="phone-form1-4t">Phone</label>

<input type="number" maxlength="10"class="form-control" name="cMob" data-form-field="Phone" id="phone-form1-4t">

</div>

</div>

</div>

<div class="row row-sm-offset">

<div class="col-md-4 multi-horizontal" data-for="Courier Weight">

<div class="form-group">

<label class="form-control-label mbr-fonts-style display-7" for="name-form1-4t">Courier Weight</label>

<input type="text" class="form-control" name="cWeight" data-form-field="Name" required="" id="name-form1-4t">

</div>

</div>

<div class="col-md-4 multi-horizontal" data-for="Courier type">

<div class="form-group">

<label class="form-control-label mbr-fonts-style display-7" for="email-form1-4t">Courier Type</label>

<input class="form-control" name="cType" data-form-field="Email" required="" id="email-form1-4t">

</div>

</div>

<div class="col-md-4 multi-horizontal" data-for="amount">

<div class="form-group">

<label class="form-control-label mbr-fonts-style display-7" for="phone-form1-4t">Amount</label>

<input class="form-control" name="cAmount" >

</div>

</div>

</div>

<div class="form-group" data-for="address">

<label class="form-control-label mbr-fonts-style display-7" for="message-form1-4t">Address</label>

<textarea type="text" class="form-control" name="cAdd" rows="7" ></textarea>

</div>

<span class="input-group-btn">

<button href="" type="submit" class="btn btn-primary btn-form display-4">Send Courier</button>

</span>

</form>

</div>

</div>

</div>

</body>

</html>

**CHAPTER 6**

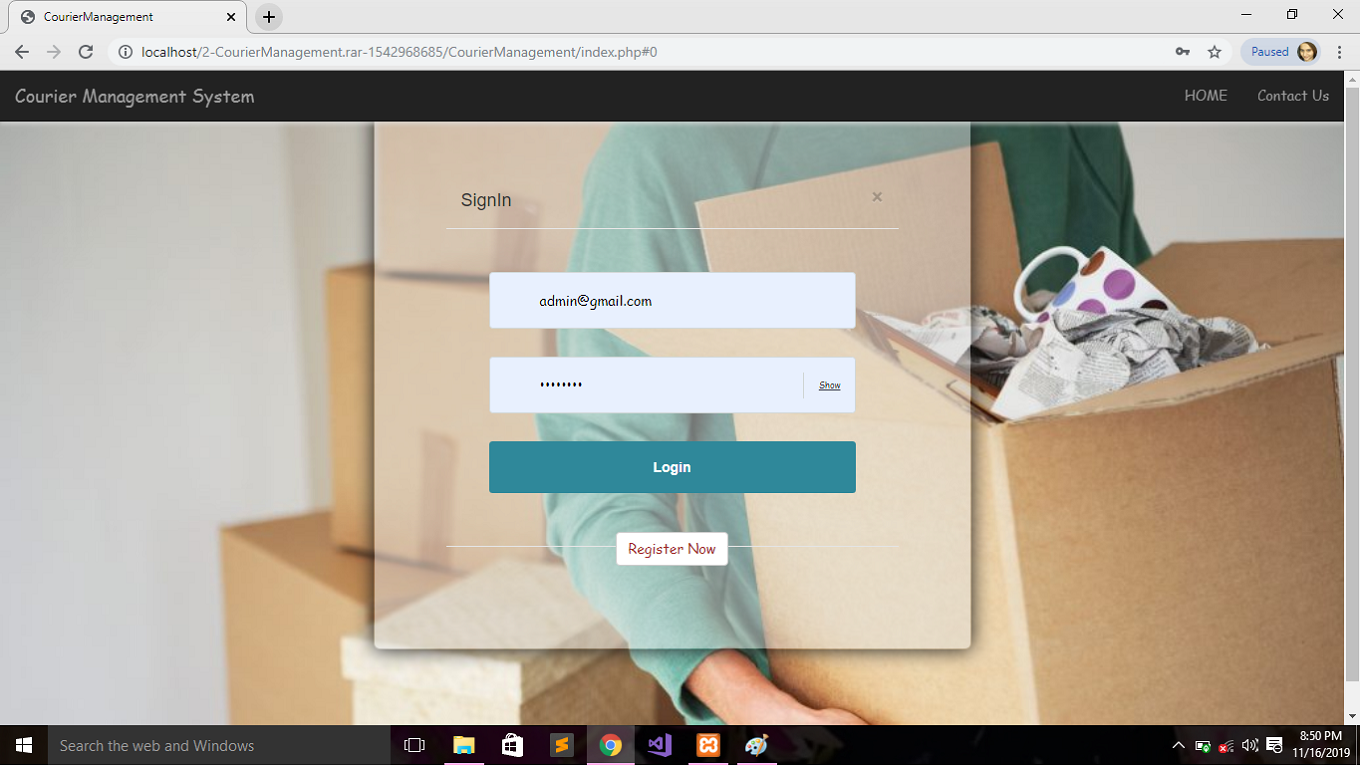
**SNAPSHOTS**

**6.1 USER REGISTRATION**

****

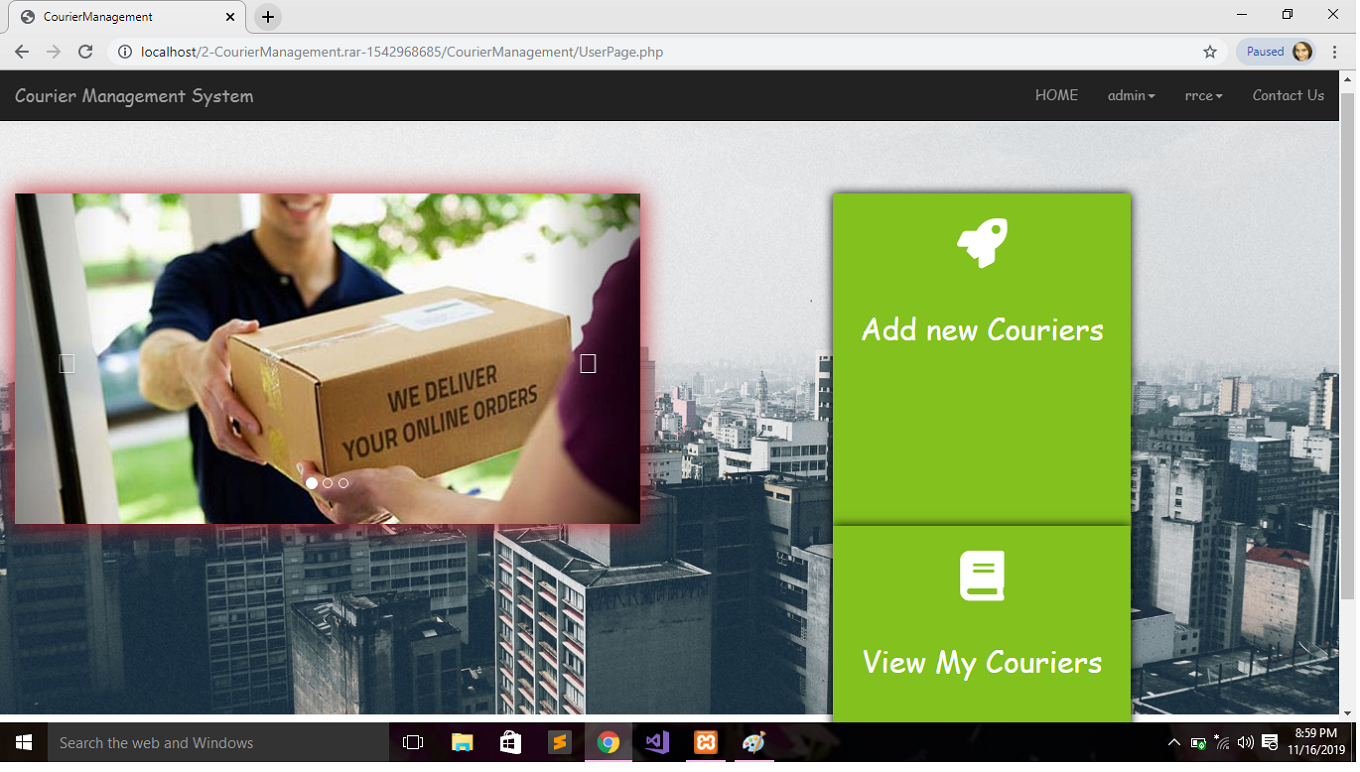
**Fig 6.1** User Registration

**6.2 USER LOGIN**



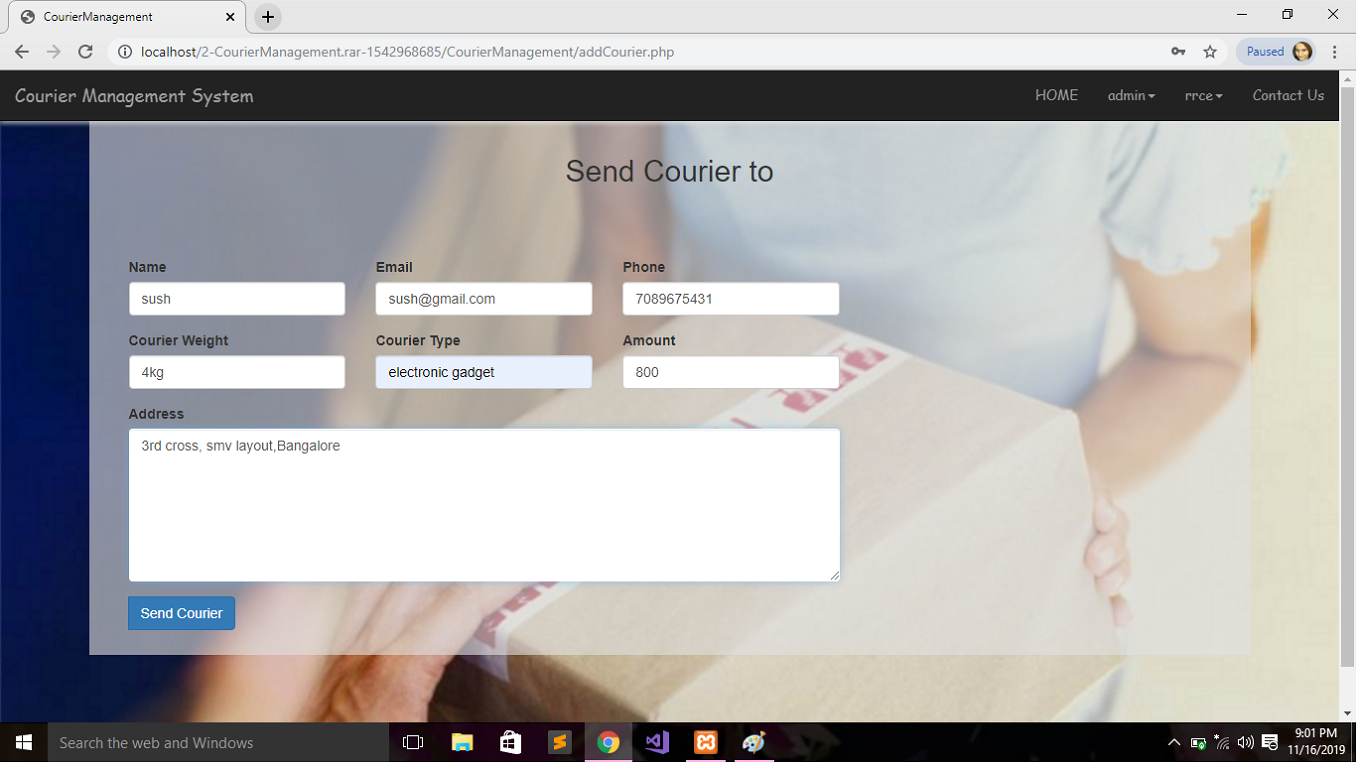
**Fig 6.2** User login

**6.3 USER PAGE**

****

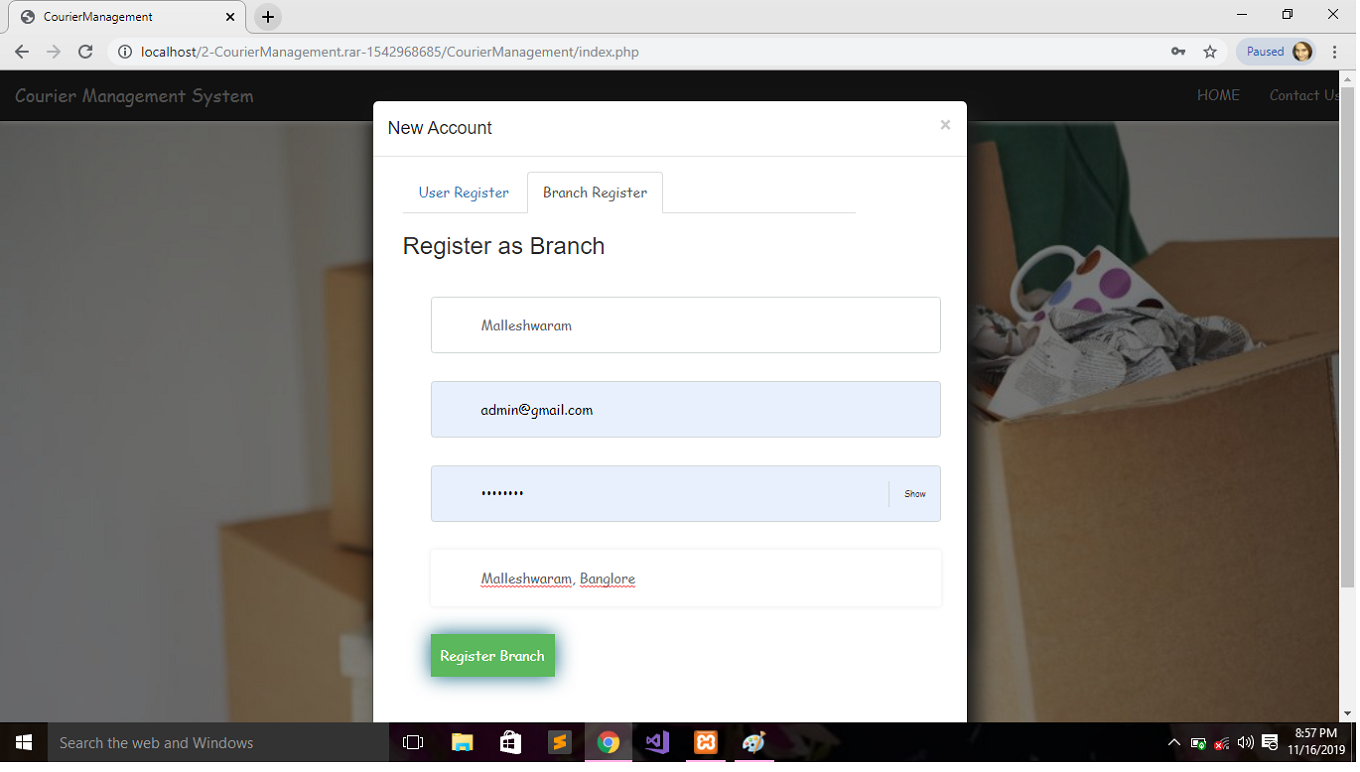
**Fig 6.3** User Page

**6.4 SEND COURIER**

****

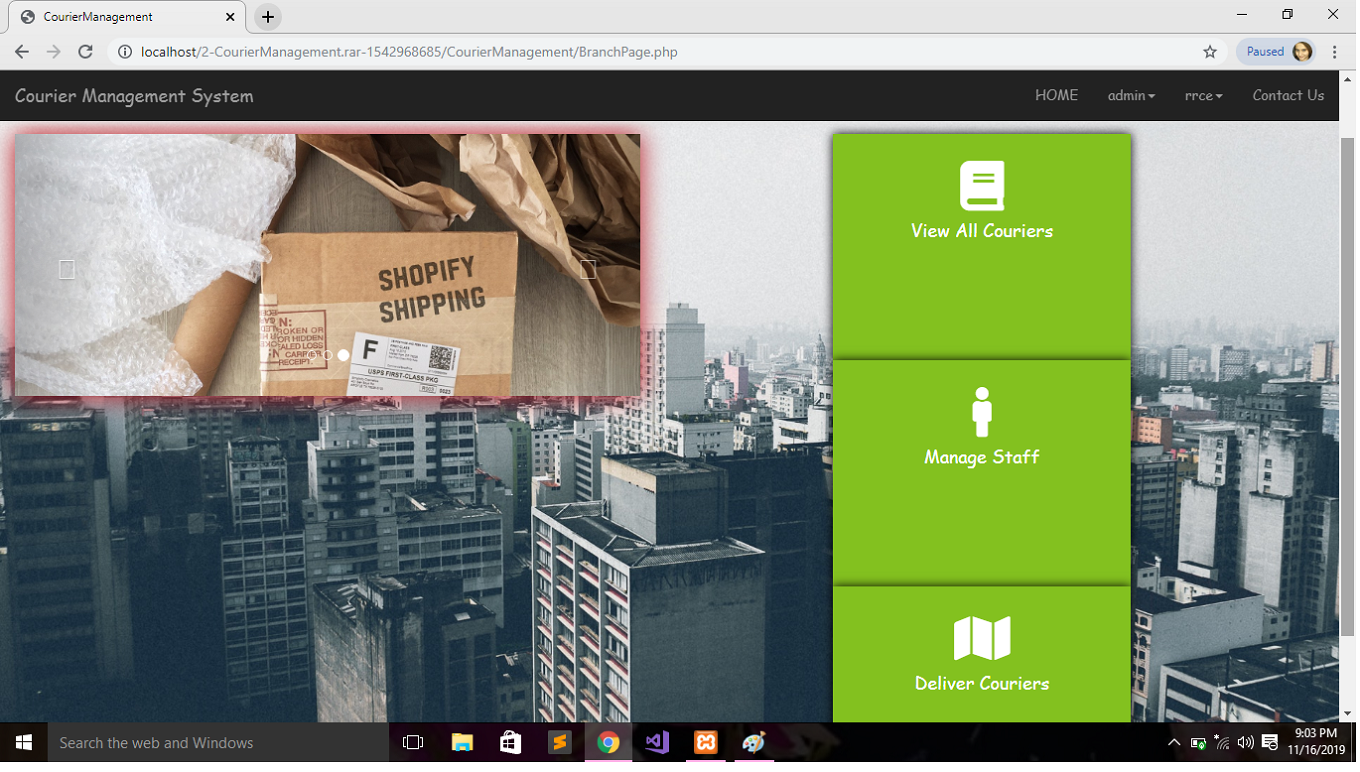
**Fig 6.4** Send Courier

**6.5 BRANCH REGISTRATION**

****

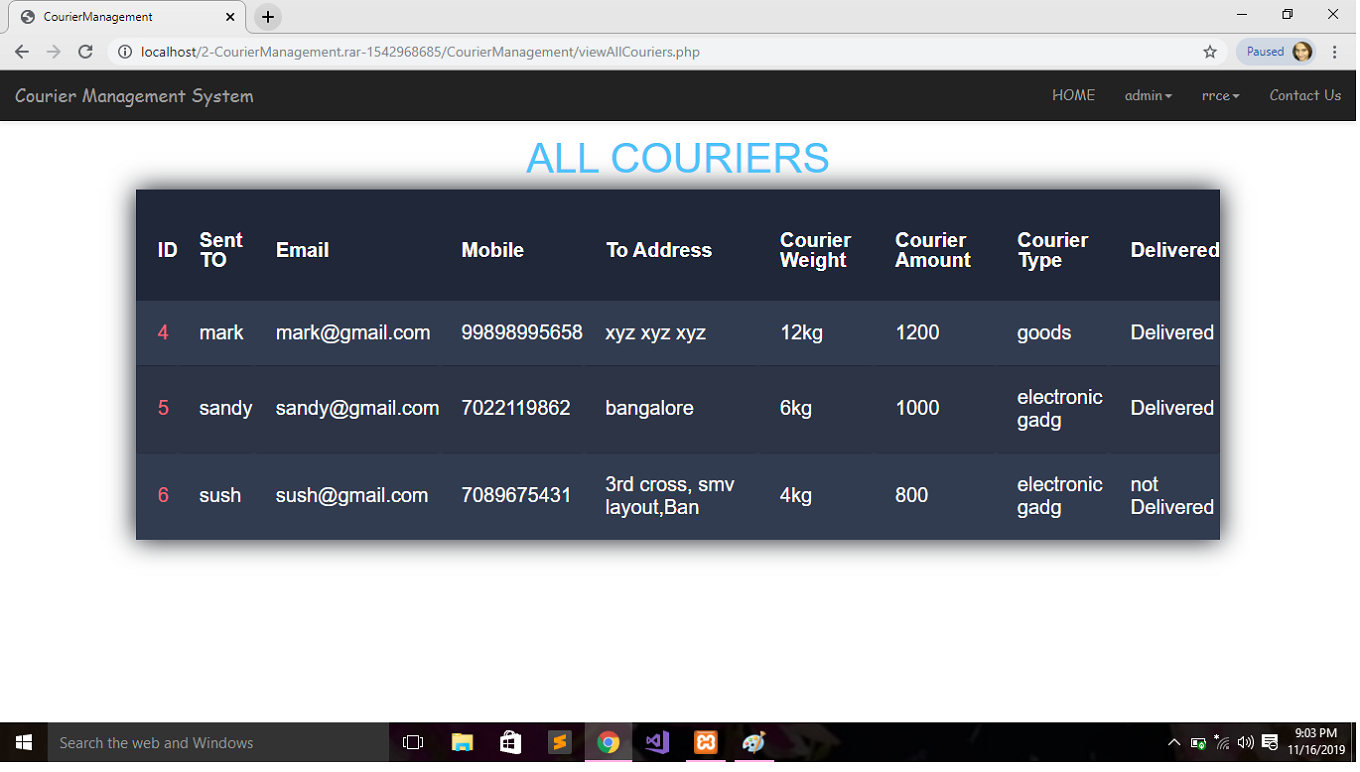
**Fig 6.5** Branch Registration

**6.6 BRANCH PAGE**

****

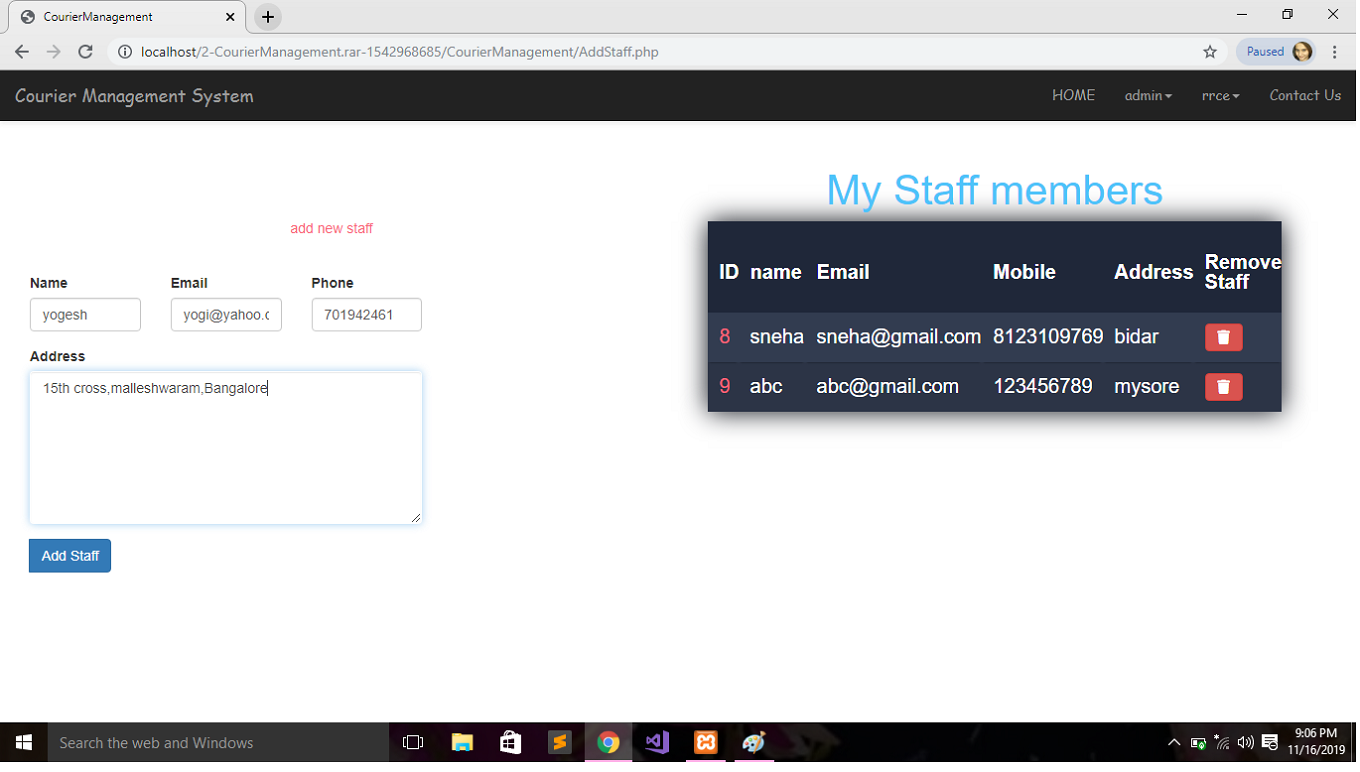
**Fig 6.6** Branch Page

**6.7 VIEW COURIERS**

****

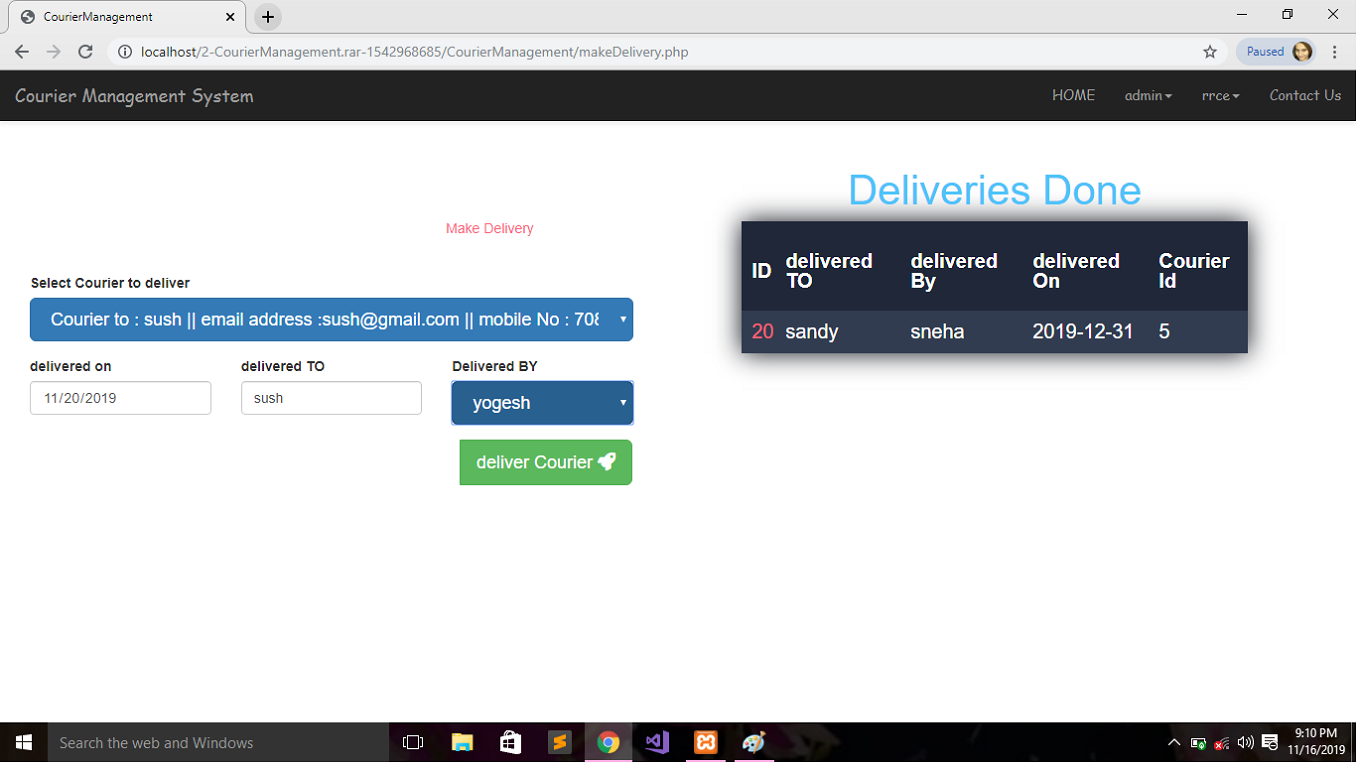
**Fig 6.7** View Couriers

**6.8 ADD STAFF**

****

**Fig 6.8** Add Staff

**6.9 MAKE DELIVERY**

****

**Fig 6.9** Make Delivery

**CONCLUSION**

After this project, User can register themselves and book for courier service using the system. And the branch manager can view and make deliveries for the couriers. Only the registered user can book for the courier and view status of the booked courier. The branch manager can add new staff members or delete them. The status of the courier can be changed by the branch manager and also specifying the staff member who delivered it.

**FUTURE ENHANCEMENT**

The use of this system is for the better services and faster processing. The application is very useful and easy to use. More time will be saved. During billing process, system generates a consignment number for their products. Through this consignment number, customer or its recipients will be able to track their products from any location using Internet.

**BIBLIOGRAPHY**

**REFERENTIAL URL’s:**

[1] [www.w3schools.com](http://www.w3schools.com)

[2] [www.tutorialspoint.com/mysql/](http://www.tutorialspoint.com/mysql/)

**REFERENCE BOOKS:**

[1] RamezElmarsi and ShamkantNavathe, “Fundamentals of Database System”, 2017 [7th Edition]

[2] Ramakrishnan and Gehrke, “Database Management System”, 2014 [3rd Edition]

[3] Bipin Desai, “An Introduction to Database System”

[4] HTML and XHTML: The Complete Reference - Thomas A. Powell

[5] PHP & MySQL Novice to Ninja – Kevin Yank