## Chapter 1

**INTRODUCTION**

This project is a Project developing a “Courier database Management System”. It is a system meant to maintain all the details regarding employees, transaction details which include courier details, customer details, goods details, retrieval of all data’s from database just by giving the delivery\_id.

The main benefit of this project is that it facilitate user to communicate in a faster manner in comparison of manual system. Through this system the current status of the courier can be known easily whereas in manual system it is a difficult task.

This software is for both cargo offices and customer. The customer can approach the office and book an article or good. The employee can use this software to enter the details of the customer and goods along with the source and destination points.

This software manages the daily transactions and also keeps the historical data in the database for future references. The system is used for daily activities such as delivery and status check. It is very difficult to do this process manually. This project is reliable only when the database is maintained properly.

This software application which avoids more manual hours that needs to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. No specific training is required for the employees to use this application. They can easily use this system to decrease manual hours spending for normal things and hence increase the performance.

The customer’s need is fulfilled by Courier Management System software which is online software for the cargo management people that enables them to receive the goods from a source and send them to a required destination and track their status from time to time. This system can be used for daily activities such as placing a courier and track the courier etc.

## Chapter 2

**SYSTEM REQUIREMENTS**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Characteristics of good SRS:

* Complete.
* Consistent.
* Feasible.
* Modifiable.
* Unambiguous.
* Testable.

## Requirements

### Software Requirements:

Operating System : Windows

User Interface : HTML, CSS

Programming Language : PHP

Database : MYSQL

### Software Features:

##### PHP

PHP is a scripting language originally designed for producing dynamic web pages.

It has evolved to include a command line interface capability and can be used in standalone graphical applications. While PHP was originally created by Rasmus Lerdorf in 1995, the main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification.

##### CSS

Cascading Style Sheets (CSS) 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 and XHTML, the language can be applied to any XML document, including plainXML, SVGand 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 webpages, 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.

### Hardware Requirements:

* Processor : INTEL corei5
* MEMORY : 1 TB
* RAM : 8 GB
* HDD : free space of 50G

## Chapter 3

**SYSTEM DESIGN**

### Tables

The existing system is a manual one in which users are maintaining ledgers to store the information like goods placing details, customer details as well as employee details. Maintaining critical information in the files and manuals Is full of risk and tedious process. The main objective of the project on Courier Management System is for the cargo management people that enables them to receive the goods from a source and send them to a required destination and for customers to track their status from time to time.

The System uses the following tables to maintain the information:

* + - Branches
    - Parcels
    - Parcel\_tracks
    - System\_settings
    - Users

The table details are as follows: -

##### Table 3.1.1: Branches

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **DEFAULT** |
| ID | INT(30) | NO | NULL |
| BRANCH\_CODE | VARCHAR(50) | NO | NULL |
| STREET | TEXT | NO | NULL |
| CITY | TEXT | NO | NULL |
| STATE | TEXT | NO | NULL |
| PIN\_CODE | VARCHAR(50) | NO | NULL |
| COUNTRY | TEXT | NO | NULL |
| CONTACT | VARCHAR(100) | NO | NULL |
| DATA\_CREATED | DATETIME | NO | CURRENT\_TIMESTAMP() |

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **DEFAULT** |
| ID | INT | NO | NULL |
| REFERENCE\_NUMBER | VARCHAR(100) | NO | NULL |
| SENDER\_NAME | TEXT | NO | NULL |
| SENDER\_ADDRESS | TEXT | NO | NULL |
| SENDER\_CONTACT | TEXT | NO | NULL |
| RECIPIENT\_NAME | TEXT | NO | NULL |
| RECIPIENT\_ADDRESS | TEXT | NO | NULL |
| RECIPIENT\_CONTACT | TEXT | NO | NULL |
| TYPE | INT(1) | NO | NULL |
| FROM\_BRANCH\_ID | INT(30) | NO | NULL |
| TO\_BRANCH\_ID | INT(30) | NO | NULL |
| WEIGHT | VARCHAR(100) | NO | NULL |
| HEIGHT | VARCHAR(100) | NO | NULL |
| WIDTH | VARCHAR(100) | NO | NULL |
| LENGTH | VARCHAR(100) | NO | NULL |
| PRICE | FLOAT | NO | NULL |
| STATUS | INT(2) | NO | NULL |
| DATA\_CREATED | DATETIME | NO | CURRENT\_TIMESTAP() |

**Table 3.1.2: Parcels**

**Table 3.1.3: Parcel\_tracks**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **DEFAULT** |
| ID | INT(30) | NO | NULL |
| PARCEL\_ID | INT(30) | NO | NULL |
| STATUS | INT(2) | NO | NULL |
| DATE\_CREATED | DATATIME | NO | CURRENT\_TIMESTAMP() |

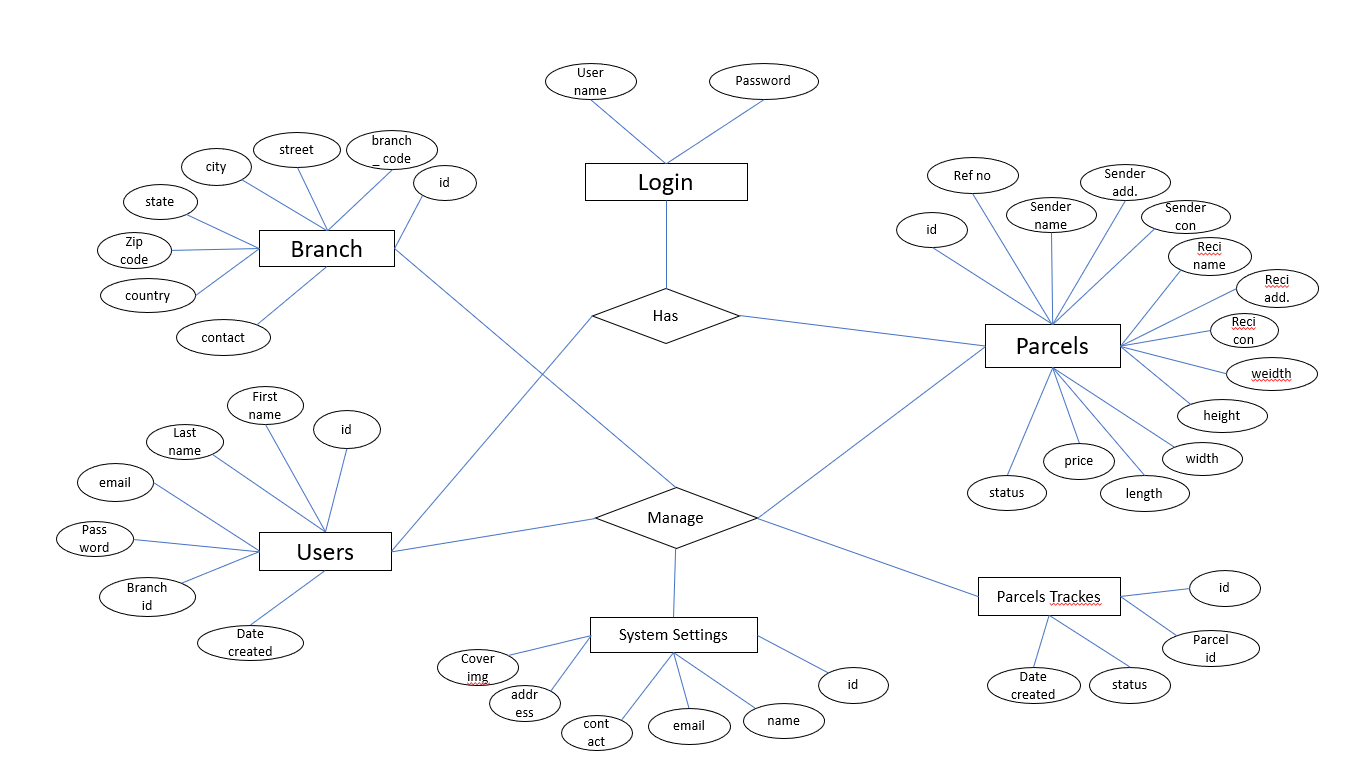
**Table 3.1.4: System\_Settings**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **DEFAULT** |
| ID | INT(30) | NO | NULL |
| NAME | TEXT | NO | NULL |
| EMAIL | VARCHAR(200) | NO | NULL |
| CONTACT | VARCHAR(200) | NO | NULL |
| ADDRESS | TEXT | NO | NULL |
| COVER\_IMG | TEXT | NO | NULL |

**Table 3.1.5: Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **DEFAULT** |
| ID | INT(30) | NO | NULL |
| FIRSTNAME | VARCHAR(200) | NO | NULL |
| LASTNAME | VARCHAR(200) | NO | NULL |
| EMAIL | VARCHAR(200) | NO | NULL |
| PASSWORD | TEXT | NO | NULL |
| TYPE | TINYINT(1) | NO | NULL |
| BRANCH\_ID | INT(30) | NO | NULL |
| DATE\_CREATED | DATETIME | NO | CURRENT\_TIMESTAMP( |

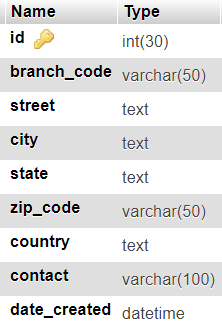
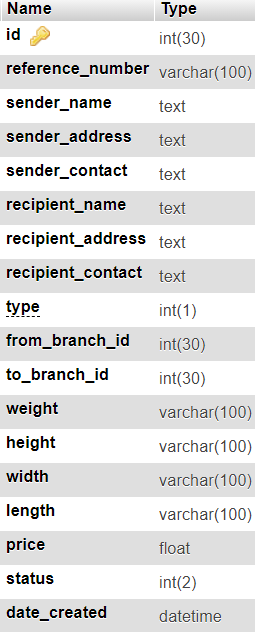
* 1. **ER DIAGRAM**: ER diagram represents the relationship between different tables.



**Figure 3.2.1:** ER Diagram for Courier Management System.

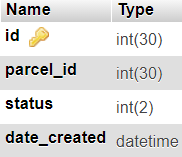
* 1. **SCHEMA DIAGRAM :**A schema is the skeleton structure that represents the logical view of the entiredatabase.

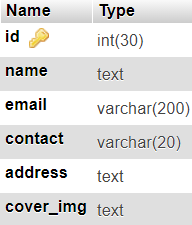




+









**Fig 3.3.1:** Schema Diagram for Courier Management System

## Chapter 4

**System Design**

Analysis is the process of breaking a complex topics or substance into smaller parts to gain a better understanding of it. Analysts in the field of engineering look at requirements, structures, mechanisms, and systems dimensions. Analysis is an exploratory activity.

Gathering requirements is the main attraction of the Analysis Phase. The process ofgatheringrequirementsisusuallymorethansimplyaskingtheuserswhattheyneedand writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements.

## Various divisions in the project

#### Home and Login

##### As Employee

1. You login as employee you can track or see all the data of parcels.
2. It allows you to update the location and also deliver the consignments to the destination.

## User Characteristics

### Administrator

* Administrator must be having good knowledge of database managements system.
* Administrator must be capable to manage user rights.
* Manual interfaces cannot be fully avoided. Documented proofs like data entry of employees etc. will have to be verified by the concerned management staff before entering it into the computerized system.

## FeasibilityStudy

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development Feasibility and risk analysis and related in many ways. If a project risk is great and feasibility of producing software is reduced. During the feasibility analysis in this project has been discussed below in the above mentioned topics.

### Operational Feasibility:

Feasibility of the working of the system after the installation in the organization as mentioned in the feasibility analysis.

### Technical Feasibility:

Technical feasibility is frequently the most difficult area to ensure this stage. It is essential that the process of analysis and definition to be conducted parallel to an assessment of the technical feasibility. The consideration that is normally associated with technical feasibility includes the resources availability of the Organization where the project is to be developed and implemented. As very limited resources are required for this project hence this project is considered feasible for development.

### Economic Feasibility:

An evaluation of development cost is weighted against the ultimate income or benefits derived from the developed system. There was no need of extra hardware and software for development of this project. Hence this project has economically justified for development in this organization.

### Motivational Feasibility:

An evaluation of the probability that the organization is sufficient motivation to support the development and implementation of the application with necessary user participation, resources, training etc. The interest and support shown by the organization during the system study do not seem that the new system developed to have efficient support from the organization.

### Schedule Feasibility:

An evaluation of the time needed for the development of this project. The time schedule required for the development of this project is very important, since more development time effects machine time, costs and delays in the development of the other systems. So the project should be complete within affixed schedule time as far as the organization is concerned.

## Chapter 5

**IMPLEMENTATION**

1. login page code for entering Username and Password:

<!DOCTYPE html>

<html lang="en">

<?php

session\_start();

include('./db\_connect.php');

ob\_start();

// if(!isset($\_SESSION['system'])){

$system = $conn->query("SELECT \* FROM system\_settings")->fetch\_array();

foreach($system as $k => $v){

$\_SESSION['system'][$k] = $v;

}

// }

ob\_end\_flush();

?>

<?php

if(isset($\_SESSION['login\_id']))

header("location:index.php?page=home");

?>

<?php include 'header.php' ?>

<body class="hold-transition login-page">

<div class="login-box">

<div class="login-logo">

<a href="#"><b><?php echo $\_SESSION['system']['name'] ?> - Admin</b></a>

</div>

<!-- /.login-logo -->

<div class="card">

<div class="card-body login-card-body">

<form action="" id="login-form">

<div class="input-group mb-3">

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

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-envelope"></span>

</div>

</div>

</div>

<div class="input-group mb-3">

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

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-lock"></span>

</div>

</div>

</div>

<div class="row">

<div class="col-8">

<div class="icheck-primary">

<input type="checkbox" id="remember">

<label for="remember">

Remember Me

</label>

</div>

</div>

<!-- /.col -->

<div class="col-4">

<button type="submit" class="btn btn-primary btn-block">Sign In</button>

</div>

<!-- /.col -->

</div>

</form>

</div>

<!-- /.login-card-body -->

</div>

</div>

<!-- /.login-box -->

<script>

$(document).ready(function(){

$('#login-form').submit(function(e){

e.preventDefault()

start\_load()

if($(this).find('.alert-danger').length > 0 )

$(this).find('.alert-danger').remove();

$.ajax({

url:'ajax.php?action=login',

method:'POST',

data:$(this).serialize(),

error:err=>{

console.log(err)

end\_load();

},

success:function(resp){

if(resp == 1){

location.href ='index.php?page=home';

}else{

$('#login-form').prepend('<div class="alert alert-danger">Username or password is incorrect.</div>')

end\_load();

}

})

})

})

</script>

<?php include 'footer.php' ?>

</body>

</html>

1. SQL queries :

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

--

-- Database: `cms\_db`

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

-- Table structure for table `branches`

--

CREATE TABLE `branches` (

`id` int(30) NOT NULL,

`branch\_code` varchar(50) NOT NULL,

`street` text NOT NULL,

`city` text NOT NULL,

`state` text NOT NULL,

`zip\_code` varchar(50) NOT NULL,

`country` text NOT NULL,

`contact` varchar(100) NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `branches`

--

INSERT INTO `branches` (`id`, `branch\_code`, `street`, `city`, `state`, `zip\_code`, `country`, `contact`, `date\_created`) VALUES

(1, 'vzTL0PqMogyOWhF', 'Branch 1 St., Quiapo', 'Manila', 'Metro Manila', '1001', 'Philippines', '+2 123 455 623', '2020-11-26 11:21:41'),

(3, 'KyIab3mYBgAX71t', 'SAmple', 'Cebu', 'Cebu', '6000', 'Philippines', '+1234567489', '2020-11-26 16:45:05'),

(4, 'dIbUK5mEh96f0Zc', 'Sample', 'Sample', 'Sample', '123456', 'Philippines', '123456', '2020-11-27 13:31:49');

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

-- Table structure for table `parcels`

--

CREATE TABLE `parcels` (

`id` int(30) NOT NULL,

`reference\_number` varchar(100) NOT NULL,

`sender\_name` text NOT NULL,

`sender\_address` text NOT NULL,

`sender\_contact` text NOT NULL,

`recipient\_name` text NOT NULL,

`recipient\_address` text NOT NULL,

`recipient\_contact` text NOT NULL,

`type` int(1) NOT NULL COMMENT '1 = Deliver, 2=Pickup',

`from\_branch\_id` int(30) NOT NULL,

`to\_branch\_id` int(30) NOT NULL,

`weight` varchar(100) NOT NULL,

`height` varchar(100) NOT NULL,

`width` varchar(100) NOT NULL,

`length` varchar(100) NOT NULL,

`price` float NOT NULL,

`status` int(2) NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `parcels`

--

INSERT INTO `parcels` (`id`, `reference\_number`, `sender\_name`, `sender\_address`, `sender\_contact`, `recipient\_name`, `recipient\_address`, `recipient\_contact`, `type`, `from\_branch\_id`, `to\_branch\_id`, `weight`, `height`, `width`, `length`, `price`, `status`, `date\_created`) VALUES

(1, '201406231415', 'John Smith', 'Sample', '+123456', 'Claire Blake', 'Sample', 'Sample', 1, 1, 0, '30kg', '12in', '12in', '15in', 2500, 7, '2020-11-26 16:15:46'),

(2, '117967400213', 'John Smith', 'Sample', '+123456', 'Claire Blake', 'Sample', 'Sample', 2, 1, 3, '30kg', '12in', '12in', '15in', 2500, 1, '2020-11-26 16:46:03'),

(3, '983186540795', 'John Smith', 'Sample', '+123456', 'Claire Blake', 'Sample', 'Sample', 2, 1, 3, '20Kg', '10in', '10in', '10in', 1500, 2, '2020-11-26 16:46:03'),

(4, '514912669061', 'Claire Blake', 'Sample', '+123456', 'John Smith', 'Sample Address', '+12345', 2, 4, 1, '23kg', '12in', '12in', '15in', 1900, 0, '2020-11-27 13:52:14'),

(5, '897856905844', 'Claire Blake', 'Sample', '+123456', 'John Smith', 'Sample Address', '+12345', 2, 4, 1, '30kg', '10in', '10in', '10in', 1450, 0, '2020-11-27 13:52:14'),

(6, '505604168988', 'John Smith', 'Sample', '+123456', 'Sample', 'Sample', '+12345', 1, 0, 0, '23kg', '12in', '12in', '15in', 2500, 1, '2020-11-27 14:06:42'),

(7, '246830784159', 'BHARATH', 'BANGALORE', '8967584930', 'UJWAL', 'KODAGU', '8967584930', 1, 1, 4, '12', '10', '23', '12', 2300, 0, '2021-01-20 12:17:19'),

(8, '801435032056', 'BHARATH A', 'BANGALORE', '8967584930', 'RAKESH', 'BELLARY', '8967584930', 1, 1, 4, '12', '10', '23', '12', 2254, 0, '2021-01-20 12:24:57');

--

-- Triggers `parcels`

--

DELIMITER $$

CREATE TRIGGER `DISCOUNT` BEFORE INSERT ON `parcels` FOR EACH ROW BEGIN SET new.price = new.price - new.price \* 0.02; END

$$

DELIMITER ;

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

-- Table structure for table `parcel\_tracks`

--

CREATE TABLE `parcel\_tracks` (

`id` int(30) NOT NULL,

`parcel\_id` int(30) NOT NULL,

`status` int(2) NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `parcel\_tracks`

--

INSERT INTO `parcel\_tracks` (`id`, `parcel\_id`, `status`, `date\_created`) VALUES

(1, 2, 1, '2020-11-27 09:53:27'),

(2, 3, 1, '2020-11-27 09:55:17'),

(3, 1, 1, '2020-11-27 10:28:01'),

(4, 1, 2, '2020-11-27 10:28:10'),

(5, 1, 3, '2020-11-27 10:28:16'),

(6, 1, 4, '2020-11-27 11:05:03'),

(7, 1, 5, '2020-11-27 11:05:17'),

(8, 1, 7, '2020-11-27 11:05:26'),

(9, 3, 2, '2020-11-27 11:05:41'),

(10, 6, 1, '2020-11-27 14:06:57');

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

--

-- Table structure for table `system\_settings`

--

CREATE TABLE `system\_settings` (

`id` int(30) NOT NULL,

`name` text NOT NULL,

`email` varchar(200) NOT NULL,

`contact` varchar(20) NOT NULL,

`address` text NOT NULL,

`cover\_img` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `system\_settings`

--

INSERT INTO `system\_settings` (`id`, `name`, `email`, `contact`, `address`, `cover\_img`) VALUES

(1, 'Courier Management System', 'info@sample.comm', '+6948 8542 623', '2102 Caldwell Road,

Rochester, New York, 14608', '')

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

--

-- Table structure for table `users`

--

CREATE TABLE `users` (

`id` int(30) NOT NULL,

`firstname` varchar(200) NOT NULL,

`lastname` varchar(200) NOT NULL,

`email` varchar(200) NOT NULL,

`password` text NOT NULL,

`type` tinyint(1) NOT NULL DEFAULT 2 COMMENT '1 = admin, 2 = staff',

`branch\_id` int(30) NOT NULL,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`id`, `firstname`, `lastname`, `email`, `password`, `type`, `branch\_id`, `date\_created`) VALUES

(1, 'Administrator', '', 'admin@admin.com', '0192023a7bbd73250516f069df18b500', 1, 0, '2020-11-26 10:57:04'),

(2, 'John', 'Smith', 'jsmith@sample.com', '1254737c076cf867dc53d60a0364f38e', 2, 1, '2020-11-26 11:52:04'),

(3, 'George', 'Wilson', 'gwilson@sample.com', 'd40242fb23c45206fadee4e2418f274f', 2, 4, '2020-11-27 13:32:12');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `branches`

--

ALTER TABLE `branches`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `parcels`

--

ALTER TABLE `parcels`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `parcel\_tracks`

--

ALTER TABLE `parcel\_tracks`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `system\_settings`

--

ALTER TABLE `system\_settings

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT for dumped tables

--

-- AUTO\_INCREMENT for table `branches`

--

ALTER TABLE `branches`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT for table `parcels`

--

ALTER TABLE `parcels`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

--

-- AUTO\_INCREMENT for table `parcel\_tracks`

--

ALTER TABLE `parcel\_tracks`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=11;

--

-- AUTO\_INCREMENT for table `system\_settings`

--

ALTER TABLE `system\_settings`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT for table `users`

--

ALTER TABLE `users`

MODIFY `id` int(30) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

COMMIT;

## Chapter 6

**Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

### Levels of Testing

##### Unit testing:

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System.by giving correct manual input to the system, the data’s are stored in database and retrieved. If you want required module to access input or get the output from the End user. Any error will accrued the time will provide handler to show what type of error will accrued.

##### System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project.it is to check all modules worked on input basis .if you want change any values or inputs will change all information so specified input I smust.

## Test cases

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:-manual and automated*.* A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical ,a few values should be chosen.

**Chapter 7**

**SCREENSHOTS**

In this chapter we will present some of the snapshots of this project.

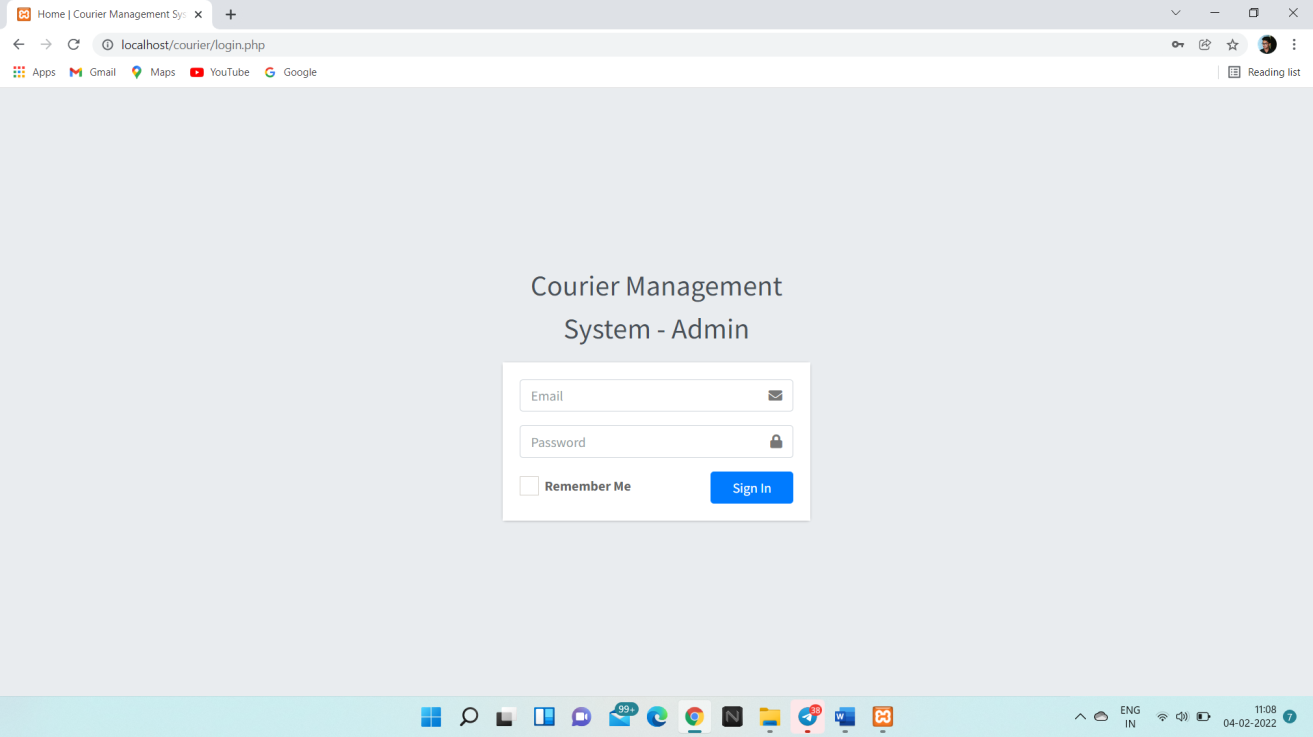
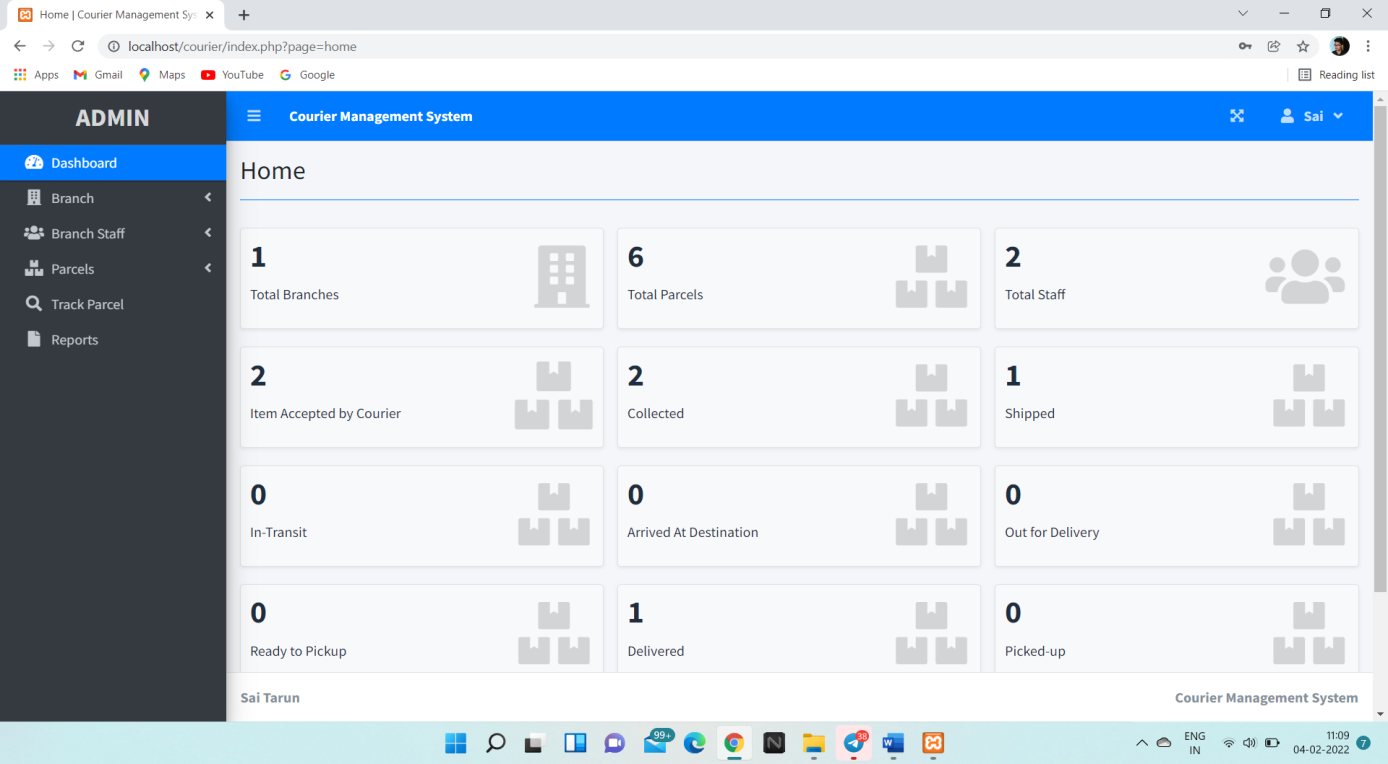
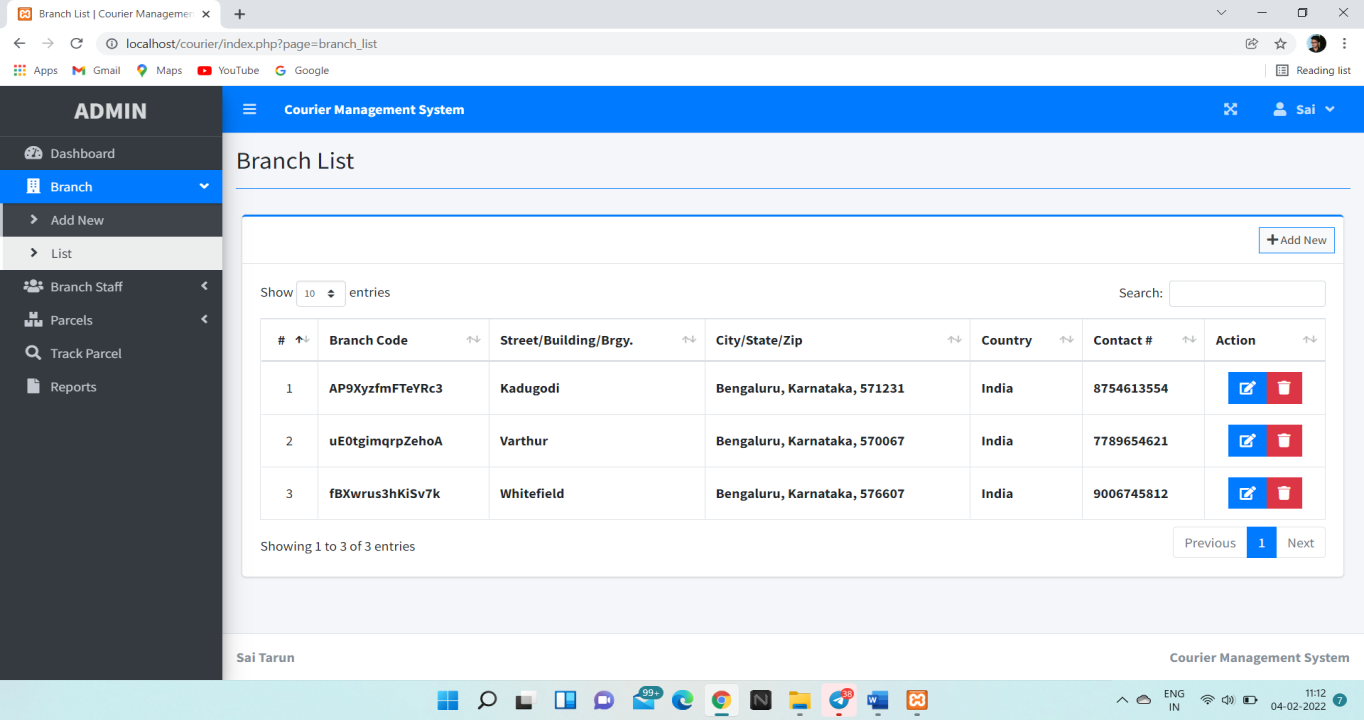


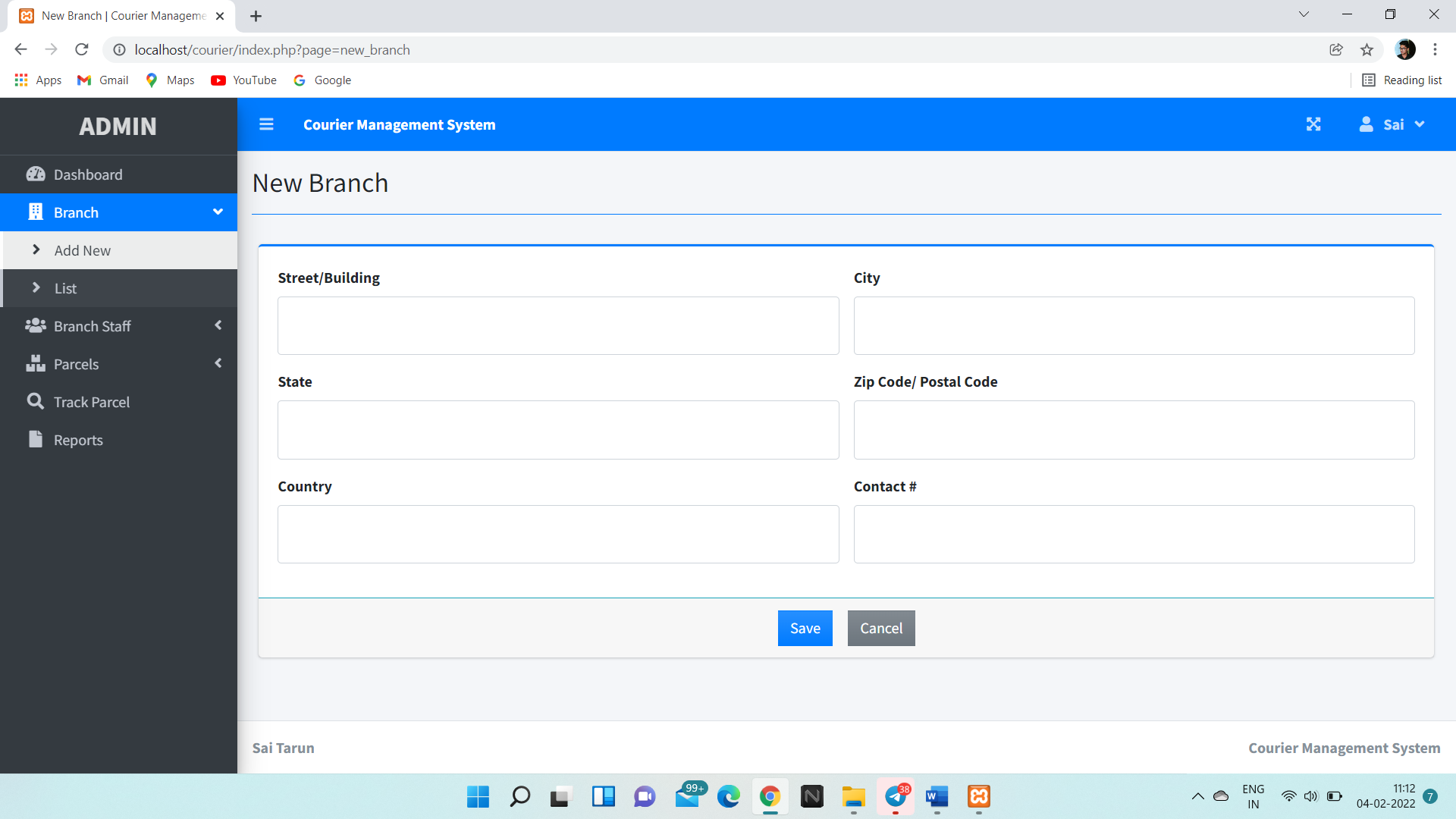
Figure 7.1: The login page.



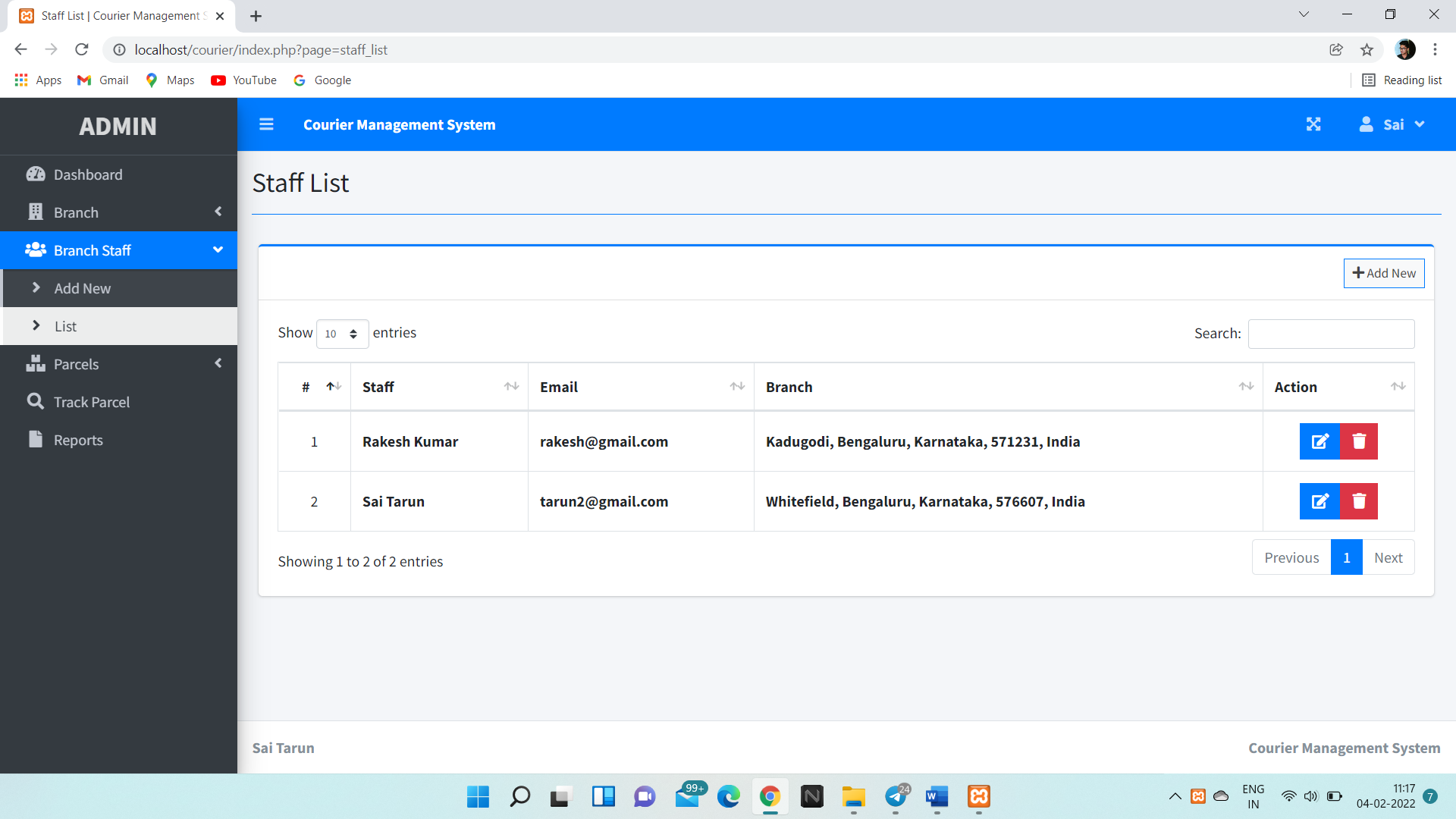
**Figure 7.2** shows the Home Page.



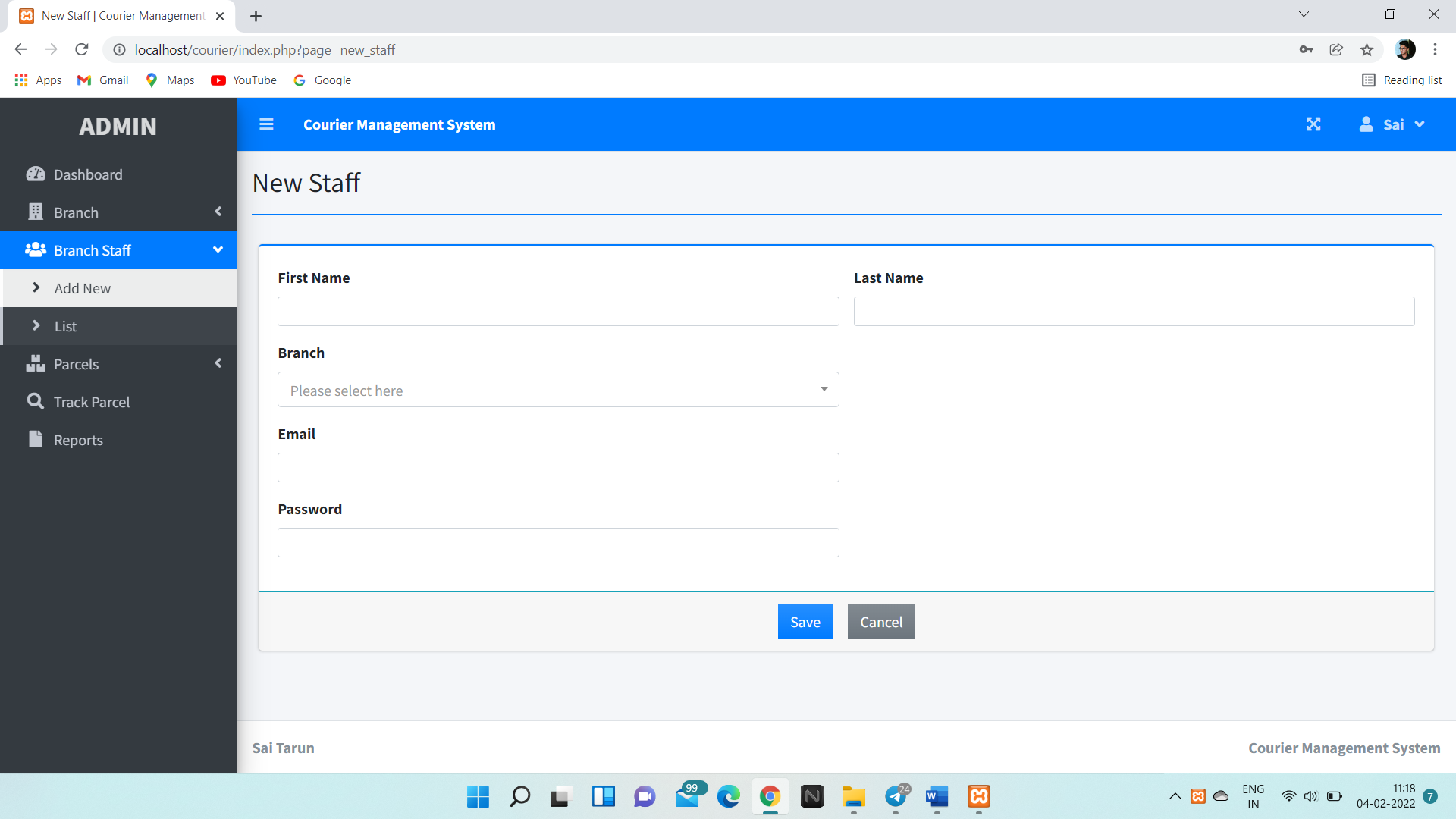
**Figure 7.3** shows the table branch list.



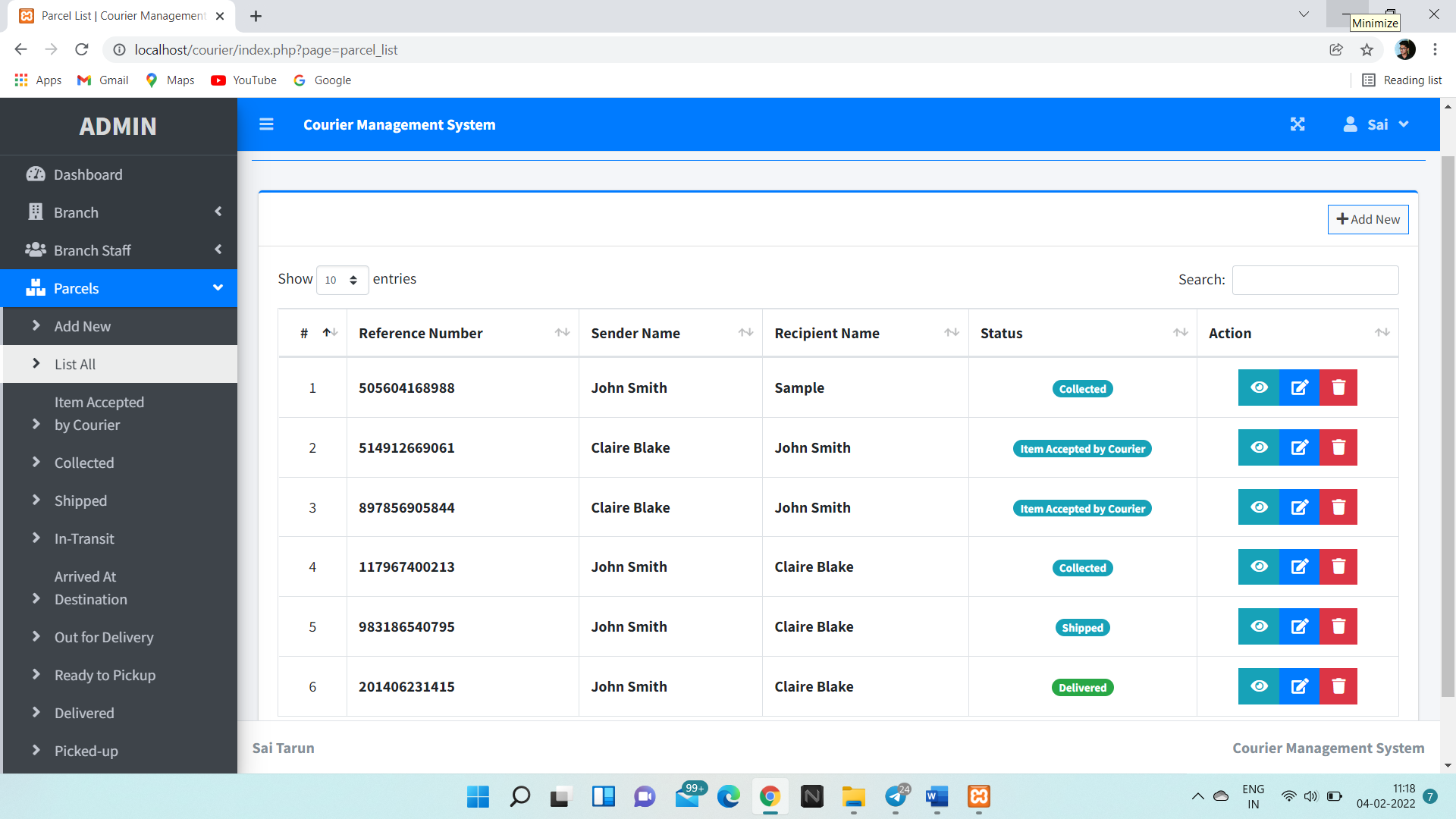
**Figure 7.4** Insert the new branch



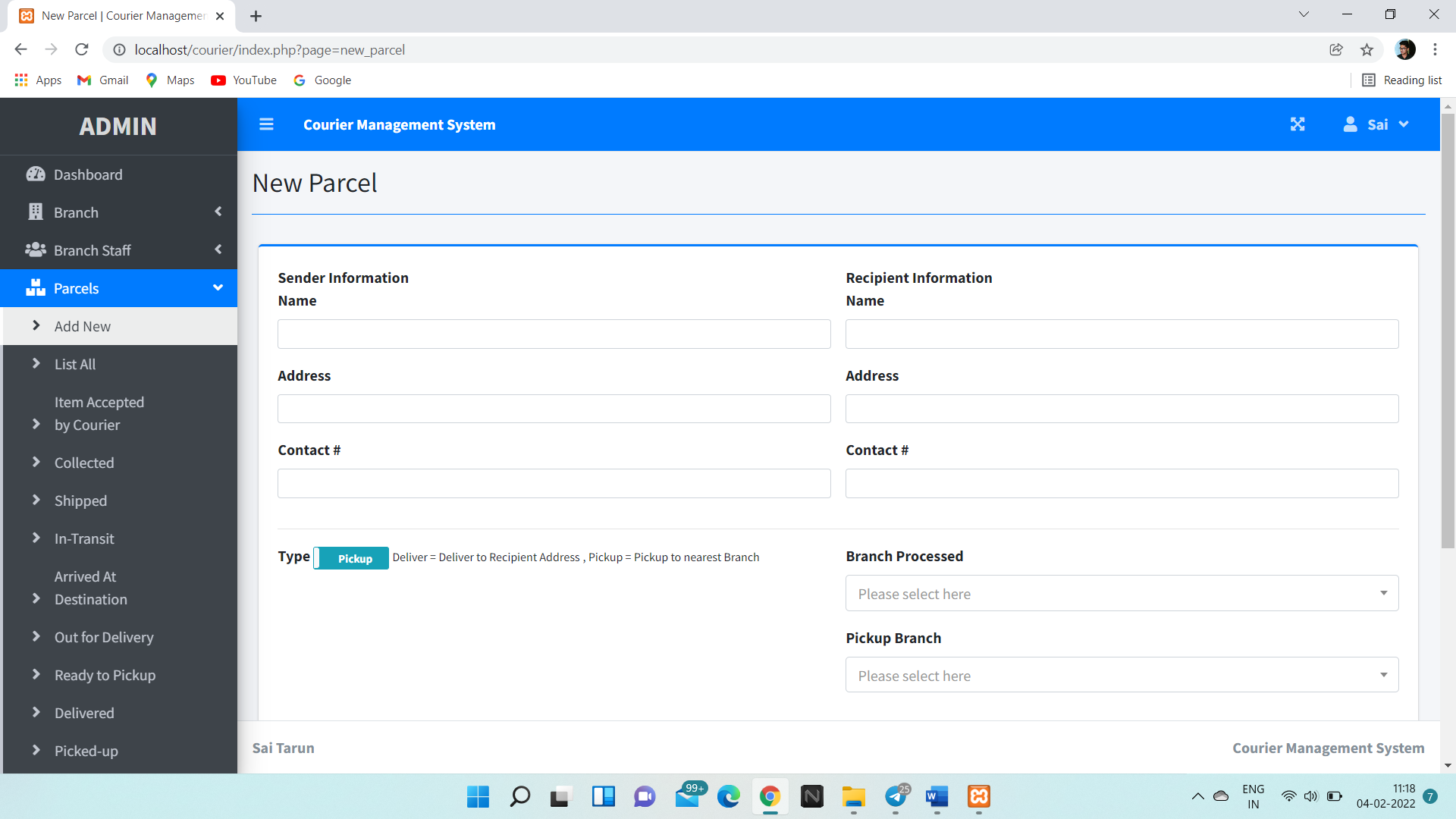
**Figure 7.5** Shows the employee or staff list



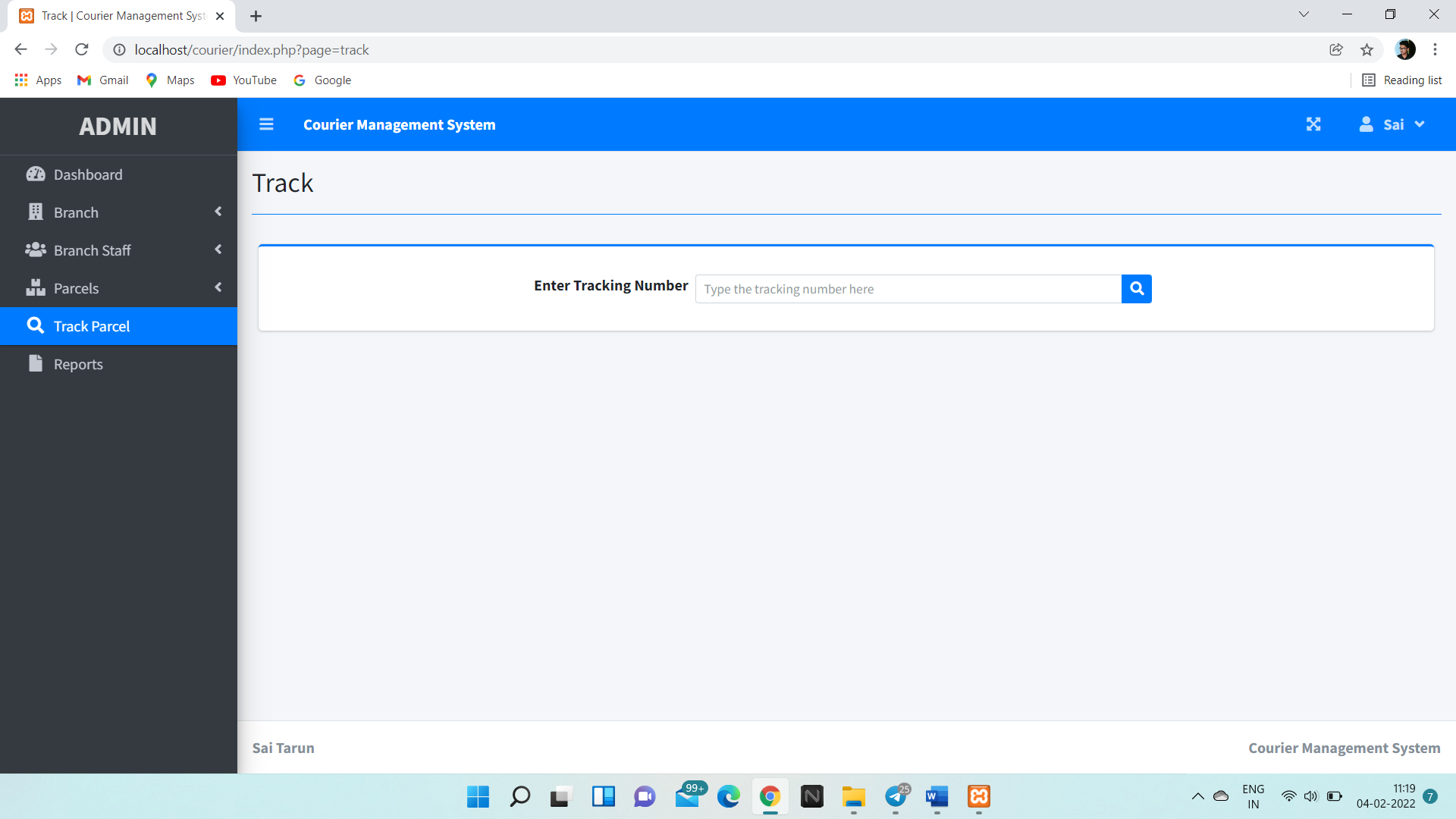
**Figure 7.6** Add new staff members



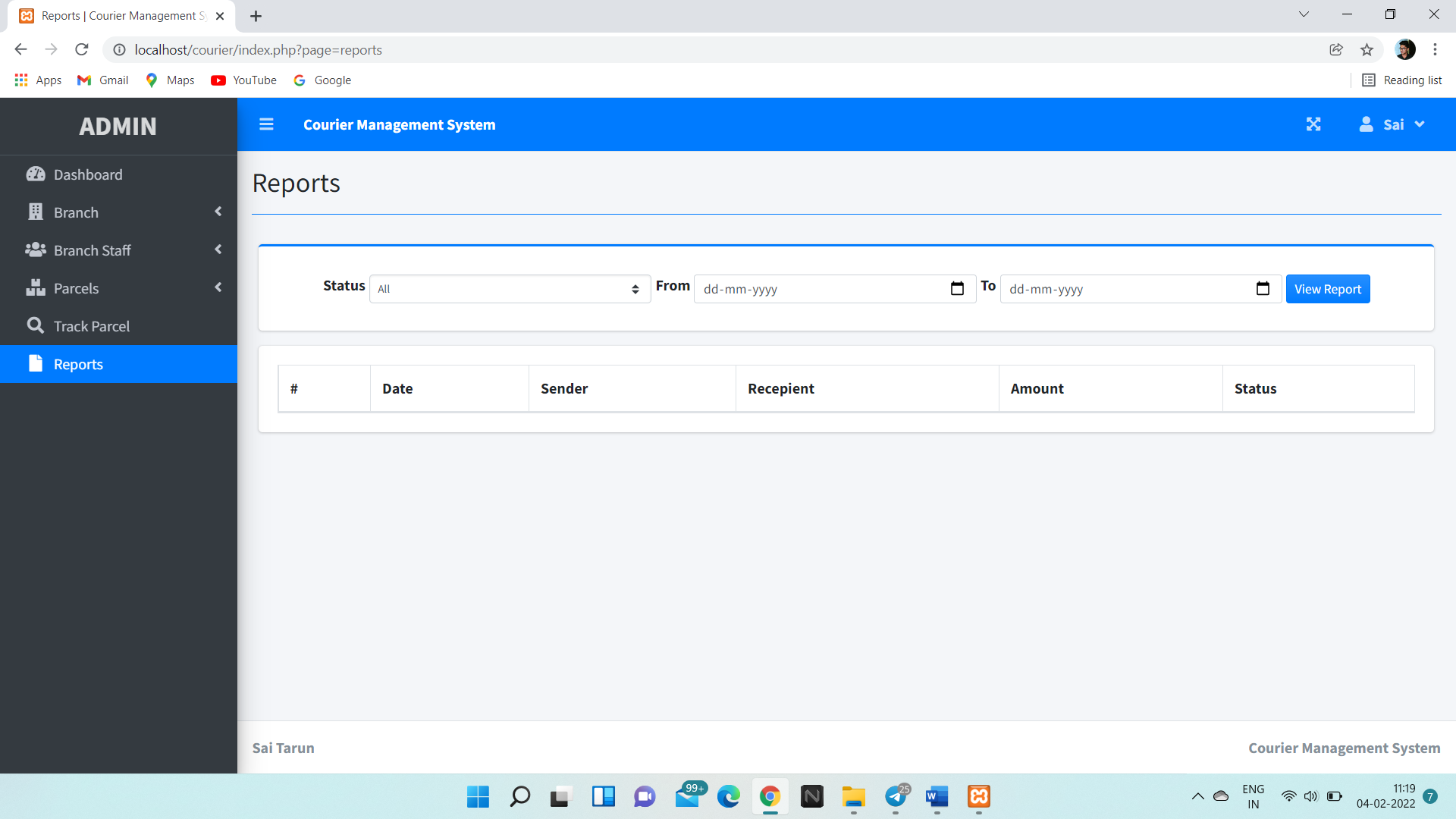
**Figure 7.7** Shows ordered Parcels list.



**Figure 7.8** Add new order or parcel.



**Figure 7.9** Tracking parcels by reference number or tracking number.



**Figure 7.10**  Reporting Page

# CONCLUSION AND FUTURE SCOPE

This project is only a humble venture to safety of customer, all parcels are tracked by employer. The project “Courier database Management System” is designed in order to avoid more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all trusted employees simultaneously. They can track the courier and track the details online with just a few clicks.

The following sections describe the future scope of the system:

* Host the platform on online servers to make it accessible worldwide
* Courier office Locator: Allows finding and choosing a nearby courier office.
* Can track all the parcels without any issues.
* Gives information to the employee about the person who is delivering the courier.
* Gives notification to the employee when the courier is reached.

# References

1. <http://www.tutorialspoint.com/Php/>
2. <http://www.tutorialspoint.com/mysql/>
3. [http://www.w3schools.com](http://www.w3schools.com/)
4. Text book Name