PROJECT REPORT

ON

COURIER MANAGEMENT SYSTEM

BY

Janhavi Hemant Gawande

Roll No: 1811017

MCA 3

Certificate from guide

Acknowledgement

I am very glad to take this opportunity to acknowledge all those who helped me in designing, developing and successful execution of my Project **"Courier Management System".**

I would like to extend my thanks and gratitude to my project guide **Mrs. Jayashree Patil** – Internal Guide for their valuable guidance and timely assistance throughout the development of this project.

I would also like to extend my thanks and gratitude to **Dr. Santosh Deshpande** (Director, IMCC**), Dr. Ravindra Vaidya** (HOD, IMCC), **Dr. Manasi Bhate** (Head – Training and Placement, IMCC) for their constant help and support.

Last but not the least, I would like to thank all the teaching and non-teaching faculties for their cooperation.

- Janhavi Gawande

INDEX

Sr No	Name Of Topic	Page No
1	**Chapter 1: Introduction**	6-20
	1.2 Existing System and Need for System	7-8
	1.3 Scope of work	9-12
	1.4 Operating Environment – Hardware and Software	13
	1.5 Detail Description of Technology Used	14-20
2	**Chapter 2: Proposed System**	21-25
	2.1 Proposed System	22-23
	2.2 Objectives Of System	24
	2.3 User Requirements	25
3	**Chapter 3: Analysis & Design**	26-75
	3.1 Object Diagram	27
	3.2 Class Diagram	28
	3.3 Use Case Diagrams	29-31
	3.4 Activity Diagrams	32-33
	3.5 Sequence Diagrams	34-35
	3.6 Entity Relationship Diagram	36
	3.7 Module Hierarchy Diagram	37
	3.8 Component Diagram	38
	3.9 Deployement Diagram	39
	3.10 Module Specifications	40-42
	3.13 User Interface Design	43-49
	3.14 Data Dictionary	50-58
	3.15 Table Specifications	59-62
	3.16 Test Procedures and Implementation	63-75

4	**Chapter 4: User Manual**	76-88
	4.1 User Manual	77-82
	4.2 Operations Manual	83
	4.3 Program Specifications	84-88
5	**Drawbacks and Limitations**	89
6	**Proposed Enhacements**	90
7	**Conclusions**	91
8	**Bibliography**	92
9	**ANNEXURES :**	93-143
	ANNEXURE 1 : USER INTERFACE SCREENS	94-116
	ANNEXURE 2 : OUTPUT REPORTS WITH DATA	117-120
	ANNEXURE 3 : SAMPLE PROGRAM CODE	121-143

CHAPTER 1

INTRODUCTION

1.2 Existing System and Need for System

The existing system is a manual one where users are maintaining ledgers, books, etc to store information like booking details, delivery particulars, details of receivers, customer information as well as employee information. It is very difficult to maintain historical data.

More manual hours are needed to generate required reports. Also it is a tedious task to manage historical data which needs more space. The customers who book their courier do not have complete details about the package as there is no tracking system used to track the package. As everything is done manually tracking of the package and informing it to the sender as well as the receiver becomes difficult.

To avoid all these problems of the existing system there was a need to develop a fully computerised system where the information is stored in servers and databases rather than manually entering the information. With the help of the computerised system users can easily track their courier, enter their details on their own through the

registration page, book a courier and manage their data efficiently.

Also the employees working for the company can also enter their own details and check details of other courier bookings done by customer and regularly update the database. This reduces the overall time in courier management process.

1.3 Scope of Work

Courier management application will be a comprehensive and complete application wherein the system will be able to manage all its operations and also manage their employees, manage delivery information, all registered consignments or packages, manage customer related information and provide necessary information to the same as notifications when he or she logs in to check the status of booked package or consignment.

The main objective of this project is to build a user friendly webs application so that the customers can easily view delivery status of the package and track their package.

The project will consist of different modules like admin, customer, employee . Each module will have its own functionality and will work accordingly.

Following are the modules and their functionalities:

1) Admin

- This Module will have the option to first save data related to the systems and then add departments in the company.
- In the registration process admin will create the logins too for the employees who can thereafter login to the application.The employees will be added according to their branches.
- Admin will generate reports and maintain the same.
- Admin will add different branches and can update delete the branches accordingly.

Business Rules:

- Only admin can add the employees according to their respective branch.
- Only he will have the authority to access all the databases related to the system.

2) Customer

- User can view the current delivery status of his parcel by tracking/reference number and also view the different branches of Courier Company.
- The customers are allowed to make partial payments for their courier as well.

Business Rules:

- Customer can send the courier only within the specified weight parameters.
- Cancellation can be done only within 24 hours of booking.
- Customers can set their delivery address only within the specified cities in the website.

3)Employee

- This module will be available for Managers and executives and other employees of the system.
- Executives once logged can update the status of delivery.
- Managers once logged in can view the status of all packages that are being managed.
- Employees will be responsible for sorting the couriers according to their type.

Business Rules:

- They can approve cancellation request made by the customer.
- Update the delivery status.

1.4 Operating Environment-Hardware and Software

Hardware Requirements:

Processor: Intel Core (i3)

Hard Disk: Minimum 1GB

RAM: 4 GB

Connectivity: Internet connection is must.

Software Requirements:

Operating System: Windows 10

Front end: HTML, CSS, Bootstrap, JavaScript, PHP.

Back end: MySQL 5.1

1.5 Detail Description of Technologies Used:

HTML:

Short form for Hyper Text Markup Language, the authoring language used to create documents on the World Wide Web. HTML is similar to SGML, although it is not a strict subset.

HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with <Html> <Head>(enter here what document is about) <Body> and ends with </Body></Html>. All the information you'd like to include in your Web page fits in between the <Body> and </Body> tags.

There are hundreds of other tags used to format and layout the information in a Web page. Tags are also used to specify hypertext links. These allow Web developers to direct users to other Web pages with only a click of the mouse on either an image or words.

The web has gone through many changes over the past few decades, but HTML has always been the fundamental language used to develop web pages. Interestingly, while websites have become more advanced and interactive, HTML has actually gotten simpler. If you compare the source of an HTML5 page with a similar page written in HTML 4.01 or

XHTML 1.0, the HTML5 page would probably contain less code. This is because modern HTML relies on Cascading Style Sheets or JavaScript to format nearly all the elements within a page.

CSS:

Short Form for Cascading Style Sheet .CSS are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in CSS, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used

to define the cell padding of table cells, the style, thickness, and colour of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate CSS.

JAVASCRIPT:

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMA Script, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate JavaScript file, which may also be viewed in a browser.

JavaScript functions can be called within <script> tags or when specific events take place. Examples include onClick, onMouseDown, onMouseUp, onKeyDown, onKeyUp, onFocus, onBlur, onSubmit, and many others. While standard JavaScript is still used for performing basic client-side functions, many web developers now prefer to use JavaScript libraries like jQuery to add more advanced dynamic elements to websites.

PHP:

Stands for Hypertext Preprocessor. PHP is a HTML embedded Web scripting language. This means PHP code can be inserted into the HTML of a Web page. When a PHP page is accessed, the PHP code is read or parsed by the server the page resides on. The output from the PHP functions on the page is typically returned as HTML code, which

can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded, users cannot view the PHP code on a page. This make PHP pages secure enough to access databases and other secure information.

A lot of the syntax of PHP is borrowed from other languages such as C, Java and Perl. However, PHP has a number of unique features and specific functions as well. The goal of the language is to allow Web developers to write dynamically generated pages quickly and easily. PHP is also great for creating database-driven Web sites.

BOOTSTRAP:

Bootstrap, or Bootstrapping, is a verb that comes from the saying, "to pull oneself up by his bootstraps." The idiom implies a person is self sufficient, not requiring help from others. Similarly, in the computing world, bootstrapping describes a process that automatically loads and executes commands.

The most fundamental form of bootstrapping is the start-up process that takes place when you start up a computer. In fact, the term BOOT as in

booting up a computer, comes from the word Bootstrap. When you turn on or restart a computer, it automatically loads a sequence of commands that initializes the system, checks for hardware, and loads the operating system. This process does not require any user input and is therefore considered a bootstrap process.

While Bootstrapping is often associated with the system boot sequence, it can be used by individual applications as well. For example, a program may automatically run a series of commands when opened. These commands may process user settings, check for updates, and load dynamic libraries, such as DLL files. They are considered Bootstrap processes because they run automatically as the program is starting up.

Bootstrap is also a popular web development framework used for creating websites. It was developed by a team at Twitter and has been an open source project since 2011. The Bootstrap framework includes CSS styles, JavaScript libraries, and HTML files. Bootstrap provides a way for developers to easily build responsive websites rather than designing them from scratch.

MYSQL:

MySQL is pronounced either "My S-Q-L" or "My Sequel," is an open source relational database management system. It is based on the Structured Query Language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a Web scripting language like PHP to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

CHAPTER 2

PROPOSED SYSTEM

2.1 Proposed System:

Courier management application will be a comprehensive and complete application wherein the system will be able to manage all its operations and also manage their employees, manage delivery information, all registered consignments or packages, manage customer related information and provide necessary information to the same when he or she visit the site to check the status of booked package or consignment.

It will allow the addition of systems necessary information through the admin panel. The system will also allow registration of employees working in Courier Services directly. It will manage various departments in system. The system will also allow adding a different type of package or consignment that Couriers deal with. It will have necessary reports to check datewise couriers and their respective status and amount earned by the company in a month.

The proposed system will allow efficient tracking of all couriers and their status effectively. The proposed system also has the functionality of barcode scanner for displaying data related to courier including tracking/reference number related to that courier.

The proposed system has the functionality of part payment as well for allowing customers to make part payments for their courier. A customer can book more than one courier at once.

Also each courier will have a unique tracking number which will be provided to the customer.

When the customer visits the website or portal he/she do not need to login for tracking their package. When user visits the website the user can directly enter their tracking number of the package and know the status of the courier as well as the dates.

2.2 Objectives of System:

The main objective of the website is to ease the process of booking and delivery of courier in more efficient way by thus reducing time. Following are some other objectives of the proposed system.

- Easy and fast retrieval of information.

- Customer details, book courier, view courier status, provide feedback, logout.

- Provide courier details like package type, weight, date of booking etc.

- Employee registration, Login, delivery management, tracking information related to couriers.

2.3 User Requirements:

The user expects that the website should provide more precise and clear information about the courier he/she is about to send.

Required functions are as follows:

- Admin sign up, login, add department employees, generate a report, view feedbacks, reply to feedbacks received, logout and delete accounts (if required).

- Customer should be able to track his courier and make part payments as well if needed.

- Provide courier details like package type, weight, date of booking etc.

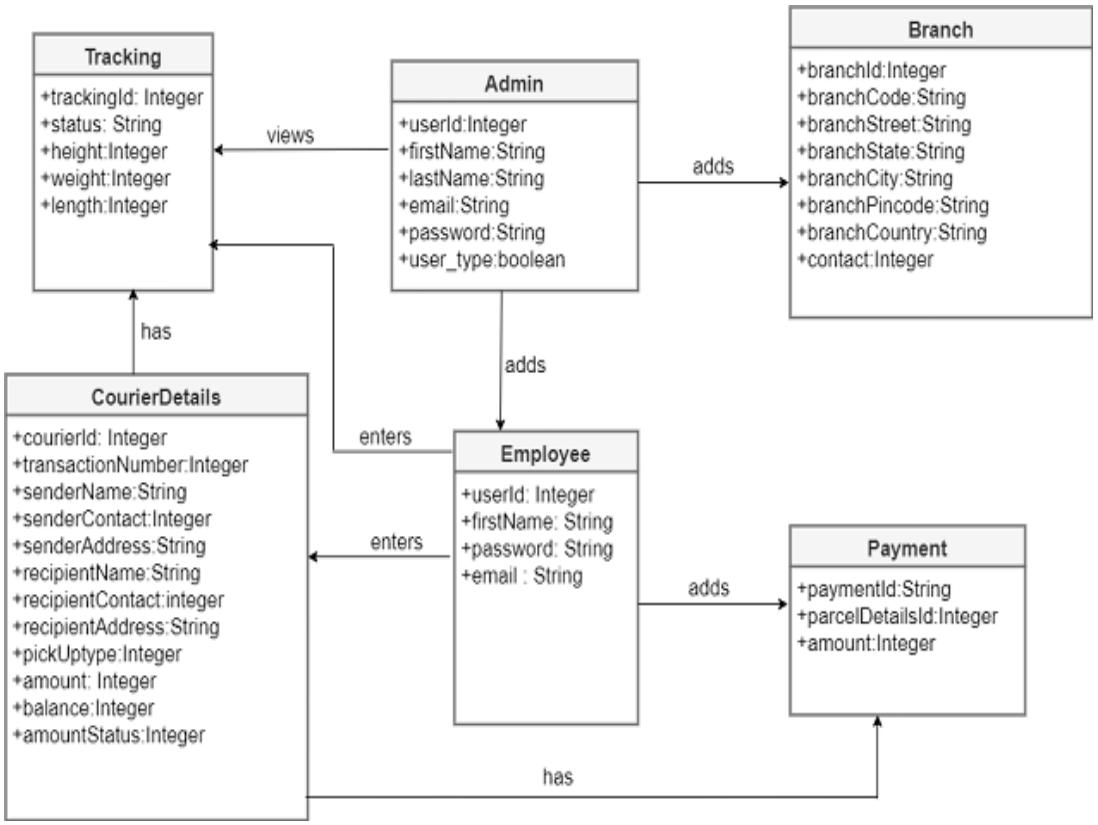
- Employee registration, Login, delivery management, tracking information related to couriers.

.

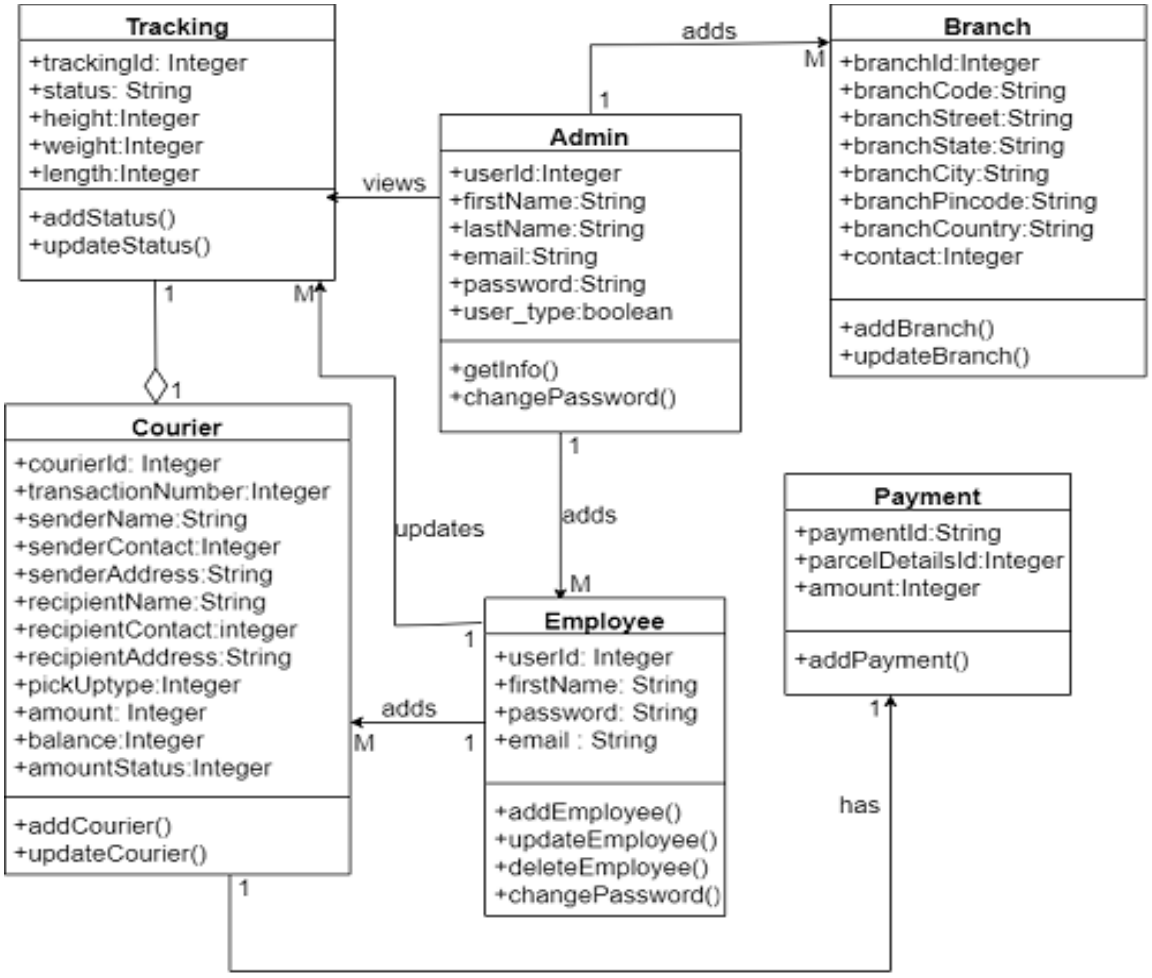
CHAPTER 3

ANALYSIS AND DESIGN

3.1 Object Diagram



3.2 Class Diagram

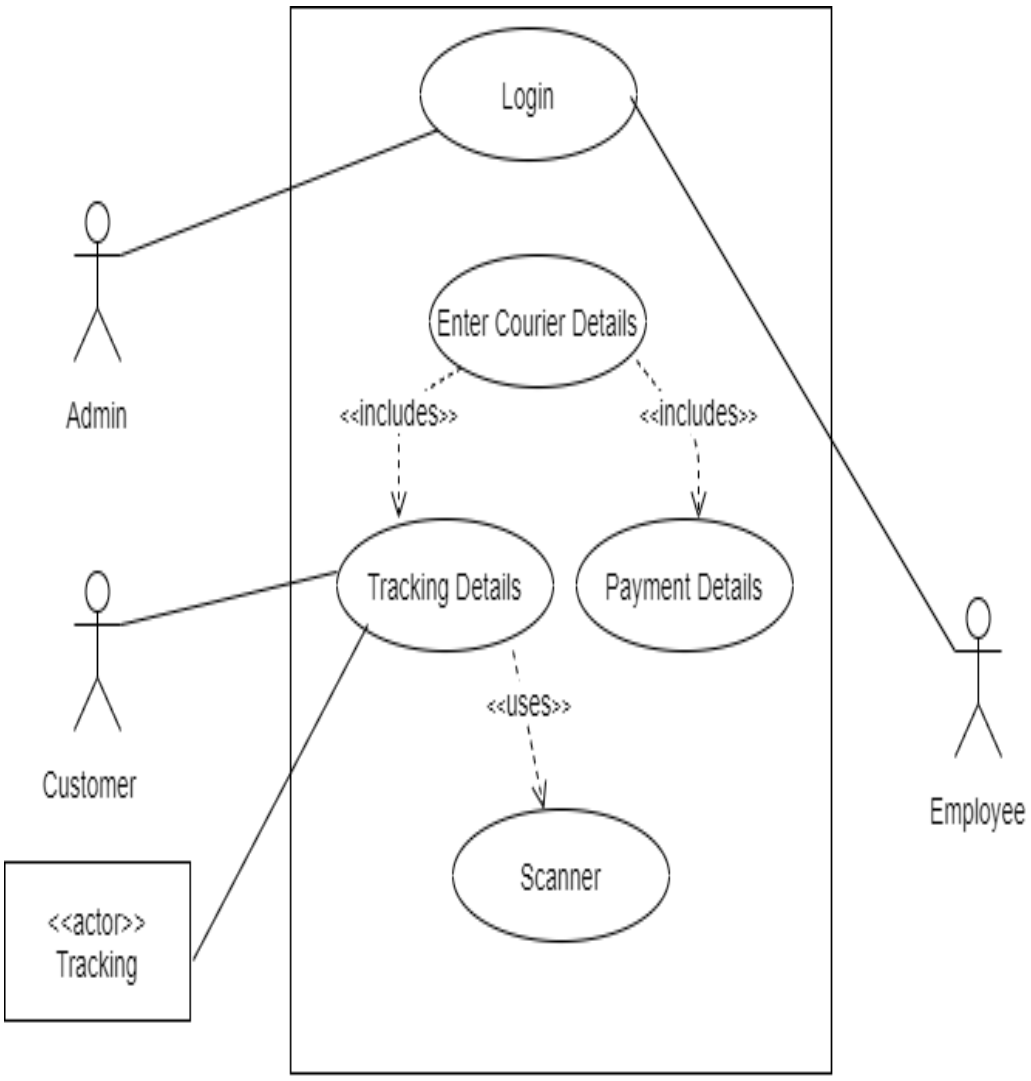


3.3 Use case diagrams

1. Complete Use Case

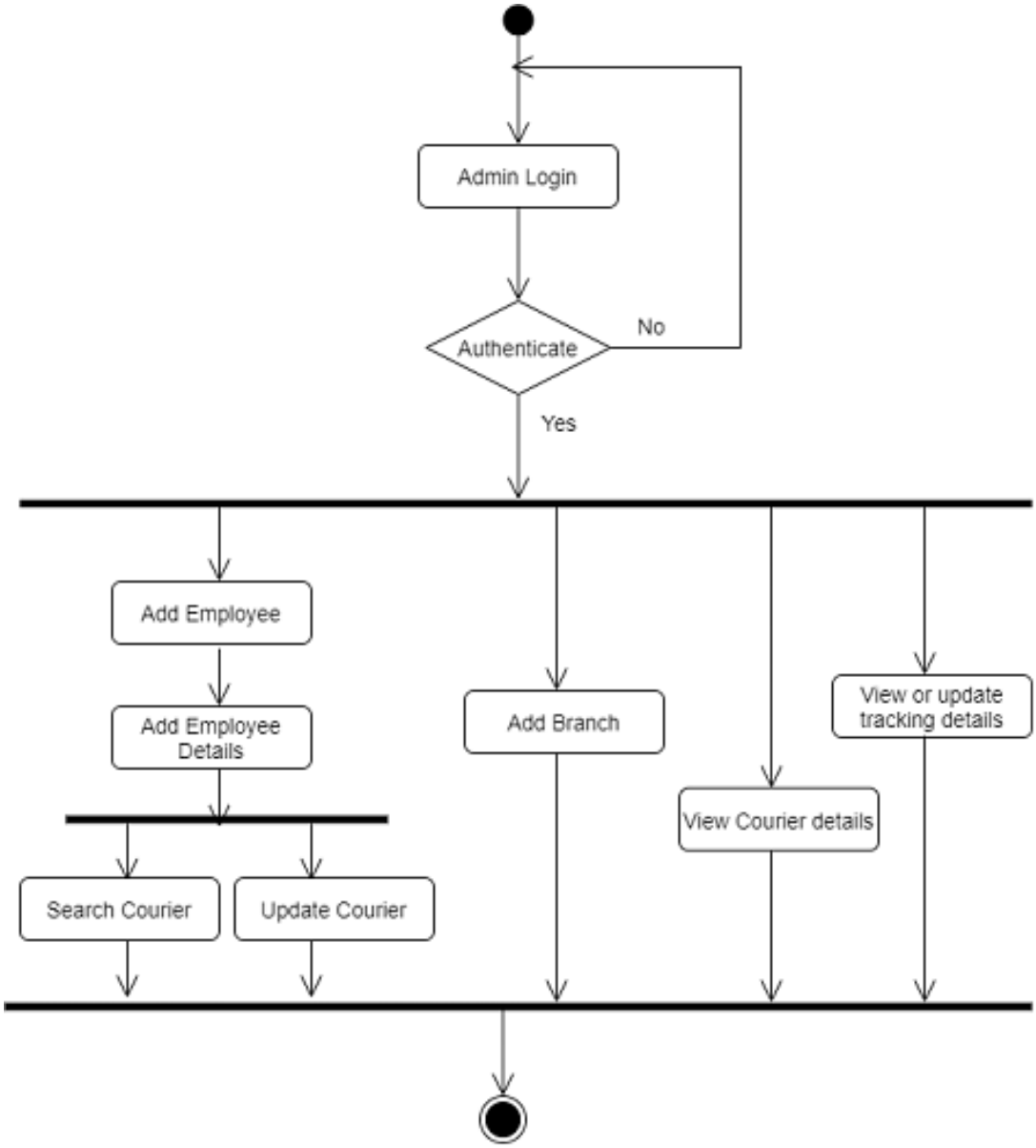
2. Admin Use Case

3.Courier Tracking Use case:

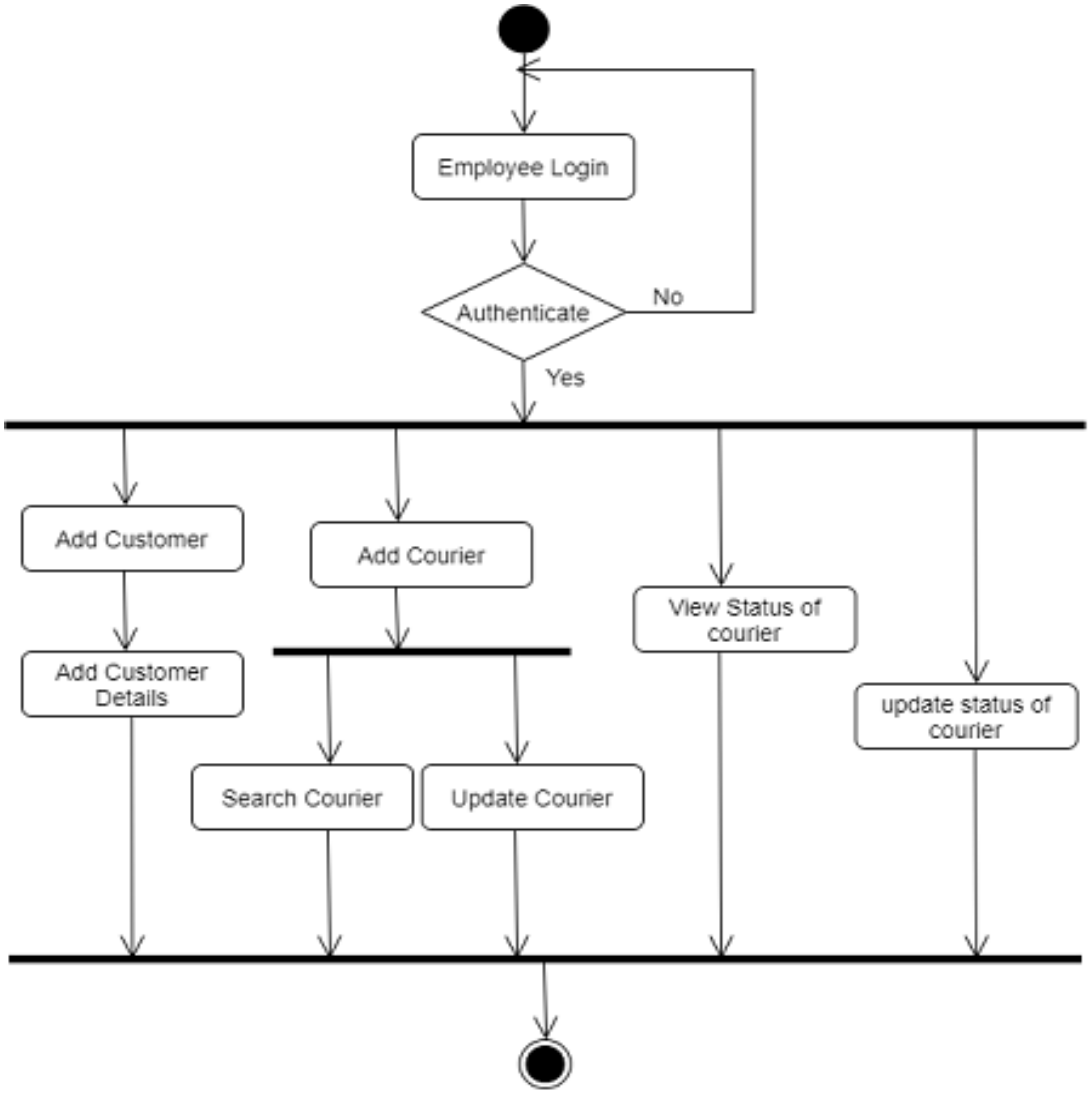


3.4 Activity Diagrams

1. Admin Activity Diagram:

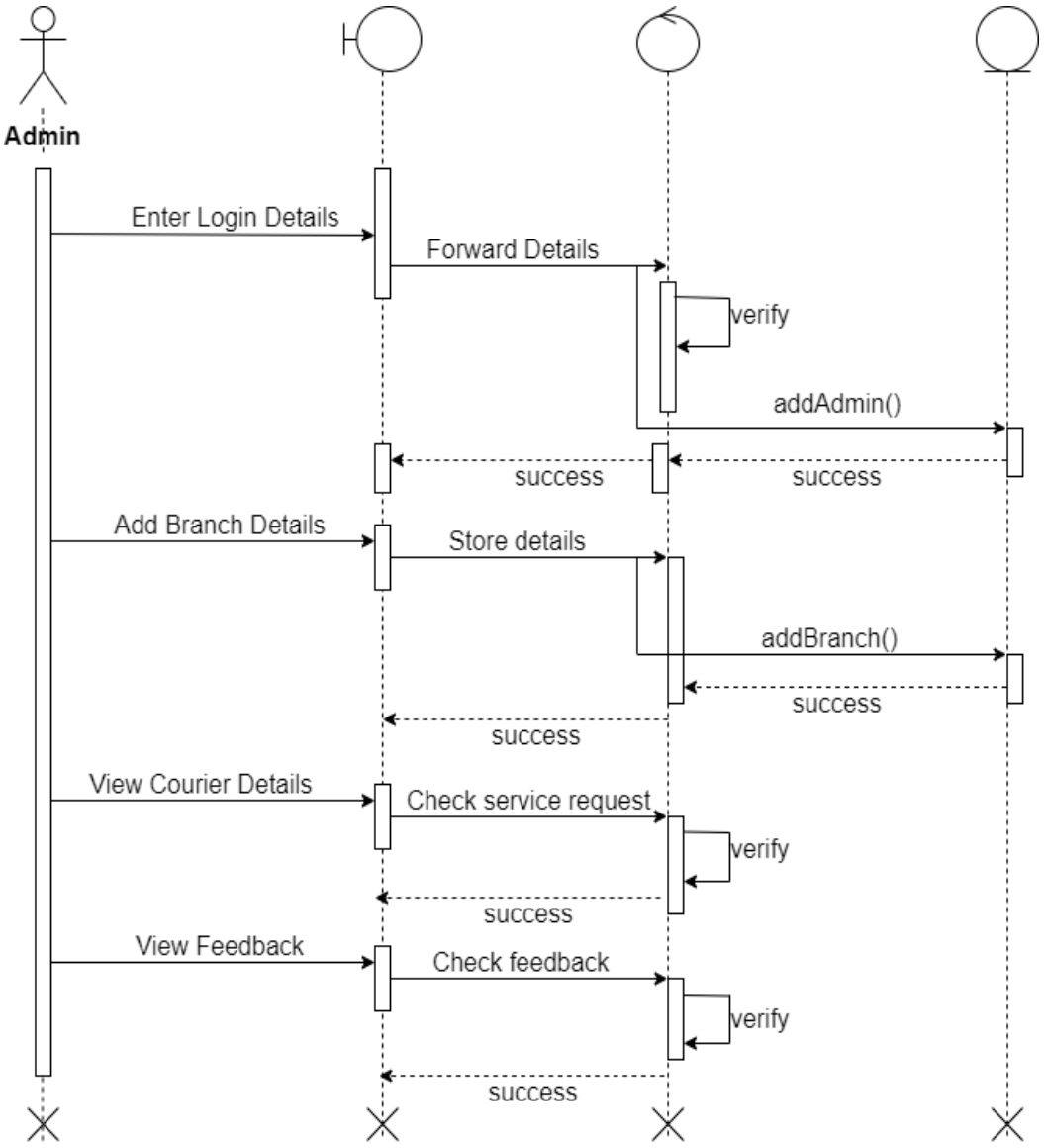


2. Employee Activity Diagram

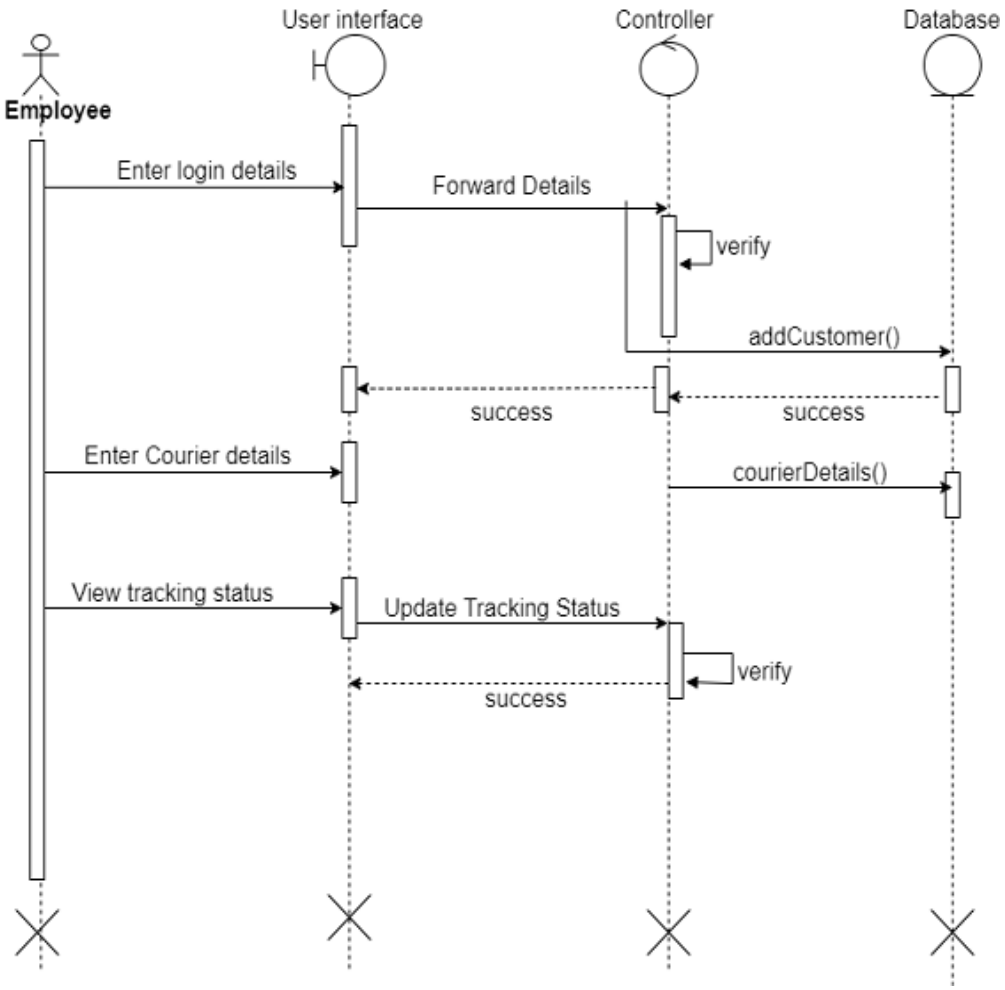


3.5 Sequence Diagram

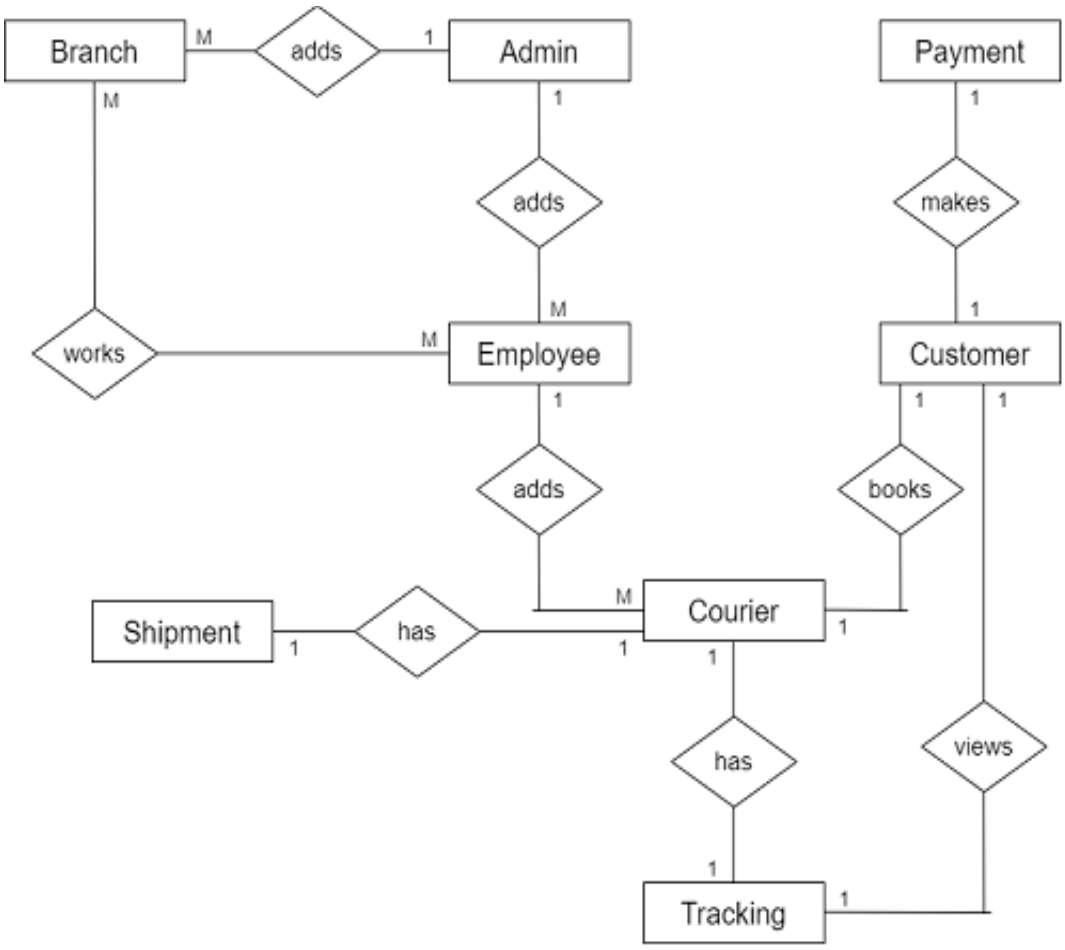
1.Admin Sequence Diagram:



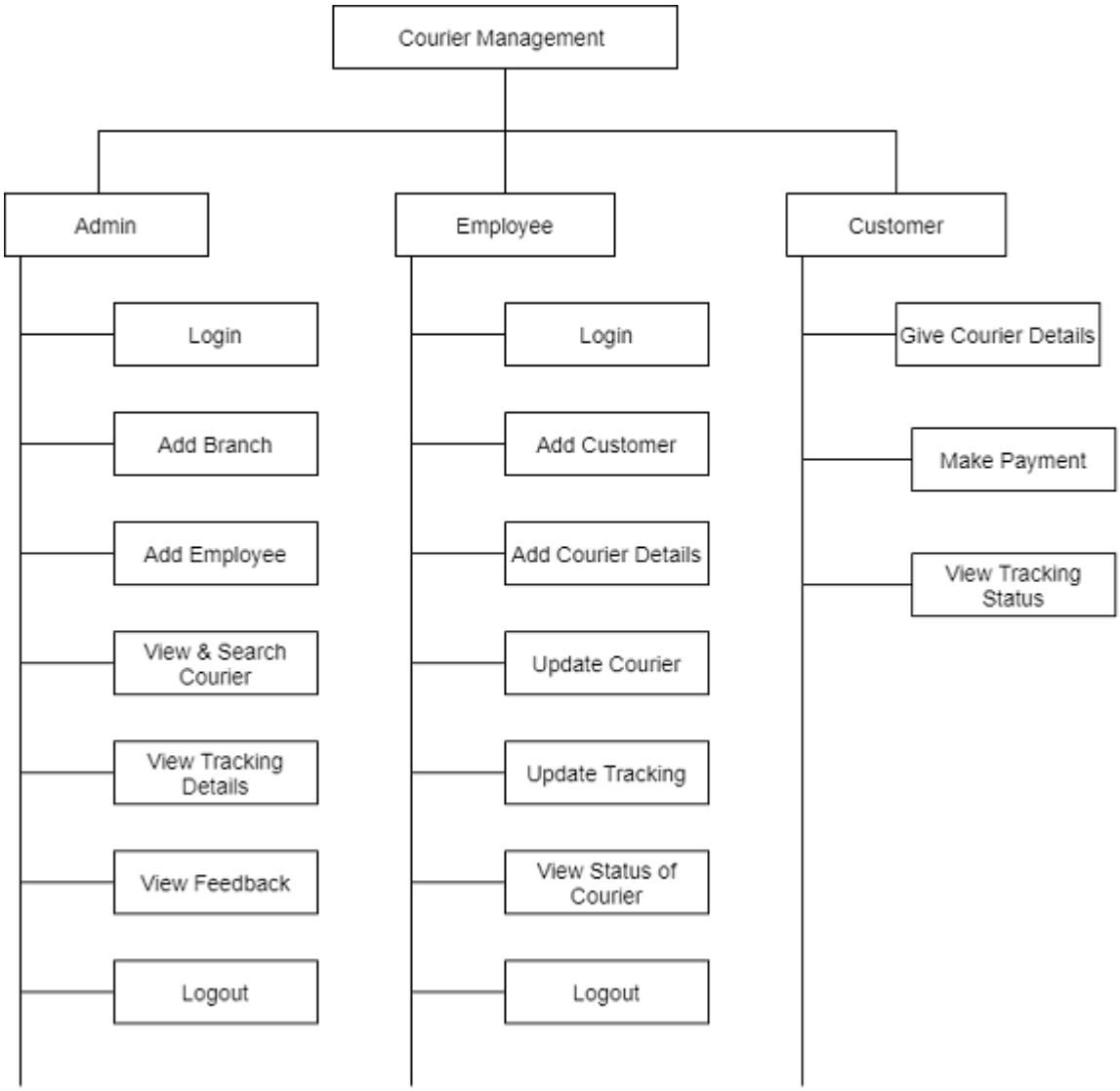
2. Employee Sequence Diagram:



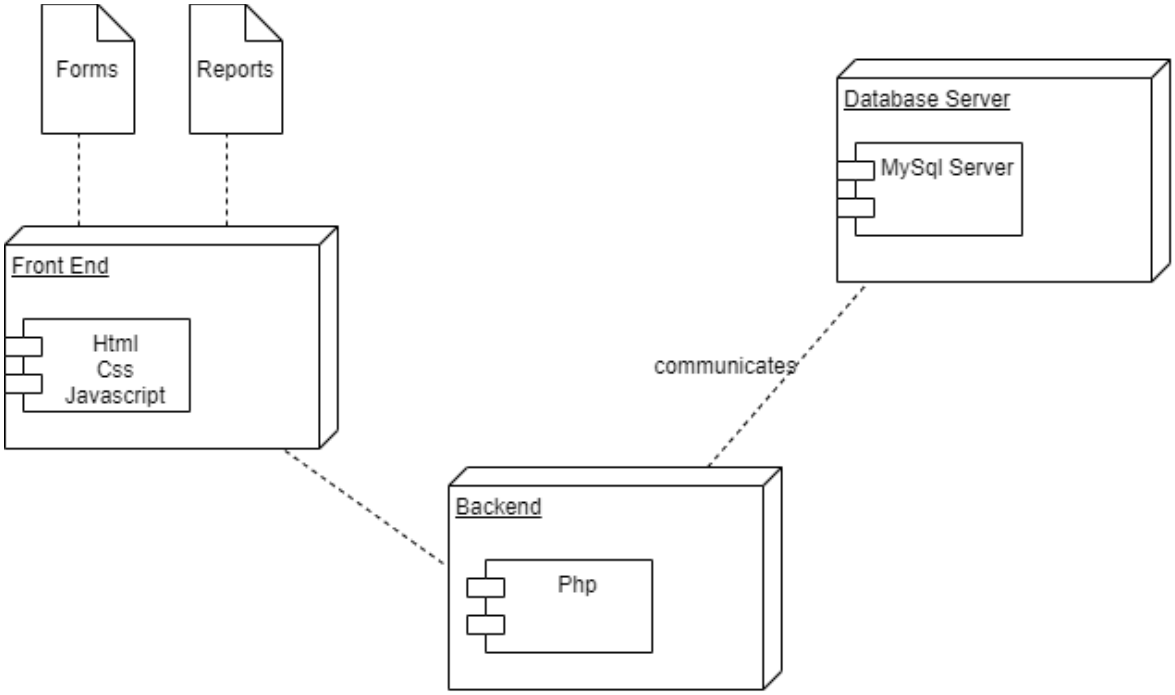
3.6 Entity Relationship Diagram



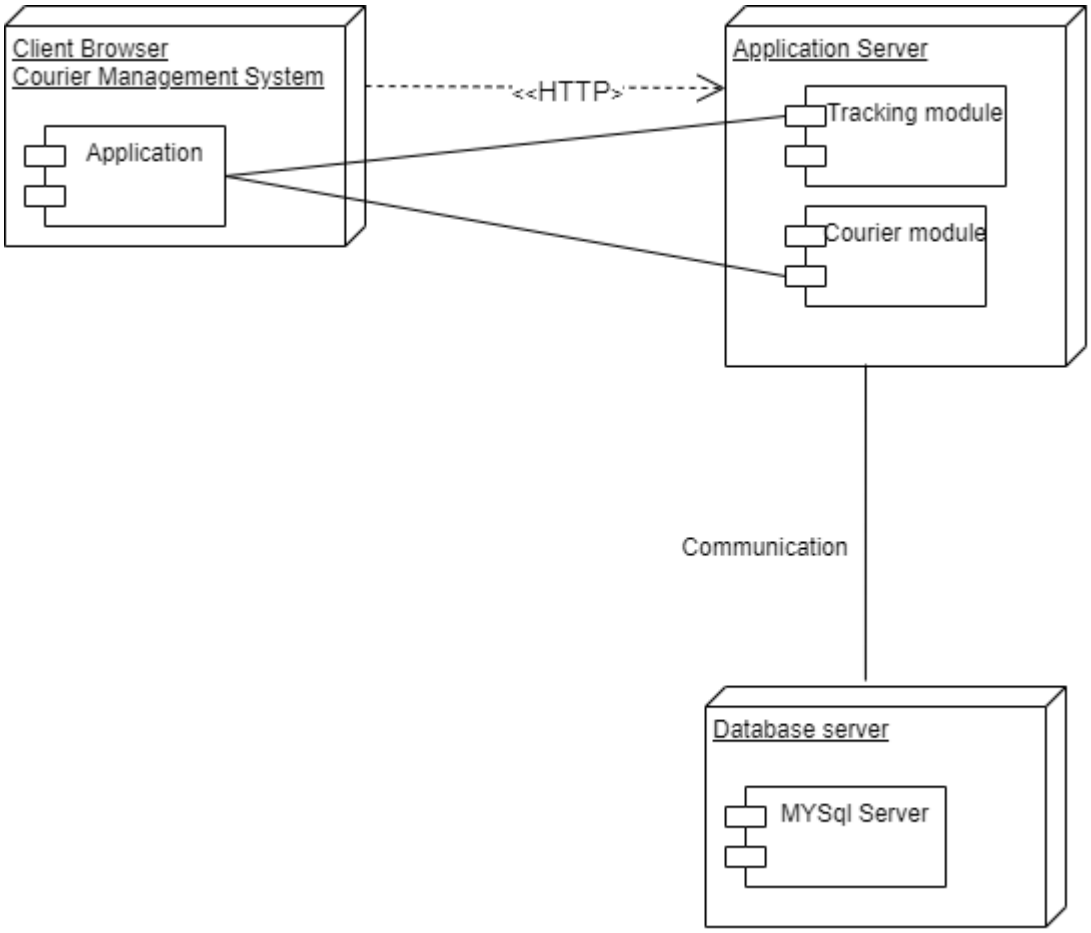
3.7 Module Hierarchy Diagram:



3.8 Component Diagram



3.9 Deployment Diagram



3.10 Module Specifications

1.Admin module

1. Dashboard: In this section, admin can see all detail in brief like total courier, Total Courier Pickup, Total Shipped, Total In-transit, Total Courier arrived at the destination, Total courier out for delivery and Total delivered courier.

2. Branches: In this section, admin can manage branches(add and update).

3. Staffs: In this section, admin can manage Staffs(add, update and delete).

4. Courier: In this section, admin can view courier status and check the courier detail which is filling by the staff of different branches.

5. Reports: In this section admin can view courier details, courier counts and sales report according to dates. Admin can also update his profile, change the password and recover the password

2.Staff Module

1. Dashboard: In this section, staffs can see all detail in brief like total courier, Total Courier Pickup, Total Shipped, Total In-transit, Total Courier arrived at the destination, Total courier out for delivery and Total delivered courier.

2. Add Courier: In this section, staffs fill the courier detail of parcel.

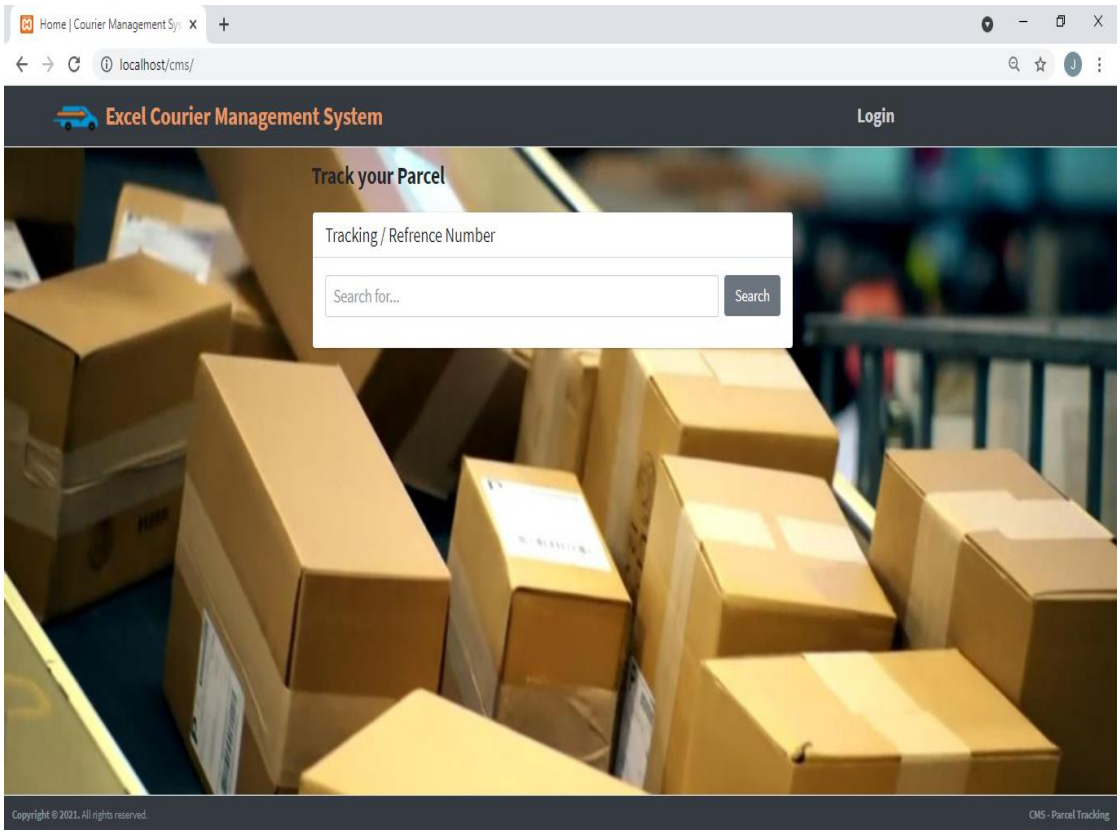
3. Status: In this section, staffs can view the courier details and them have also right to change courier status according to current status.

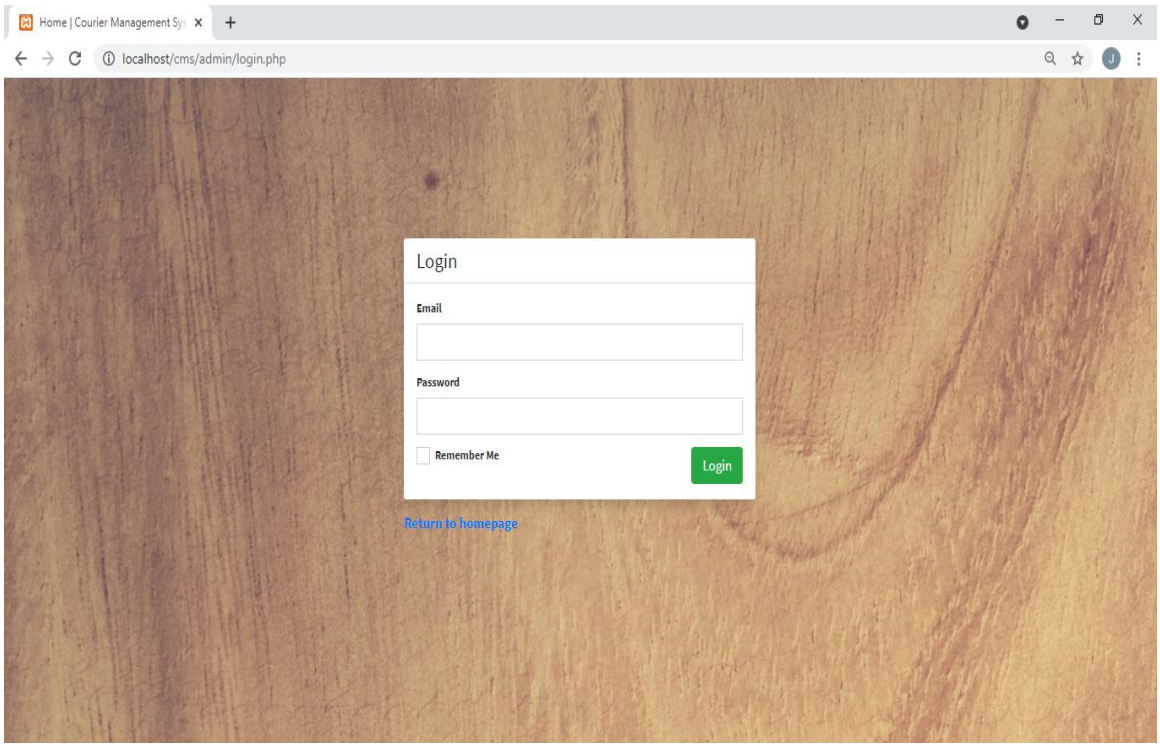
4. Search Courier: In this section, staffs can search particular courier with the help of tracking number/reference number. Staffs can also update his profile, change the password and recover the password.

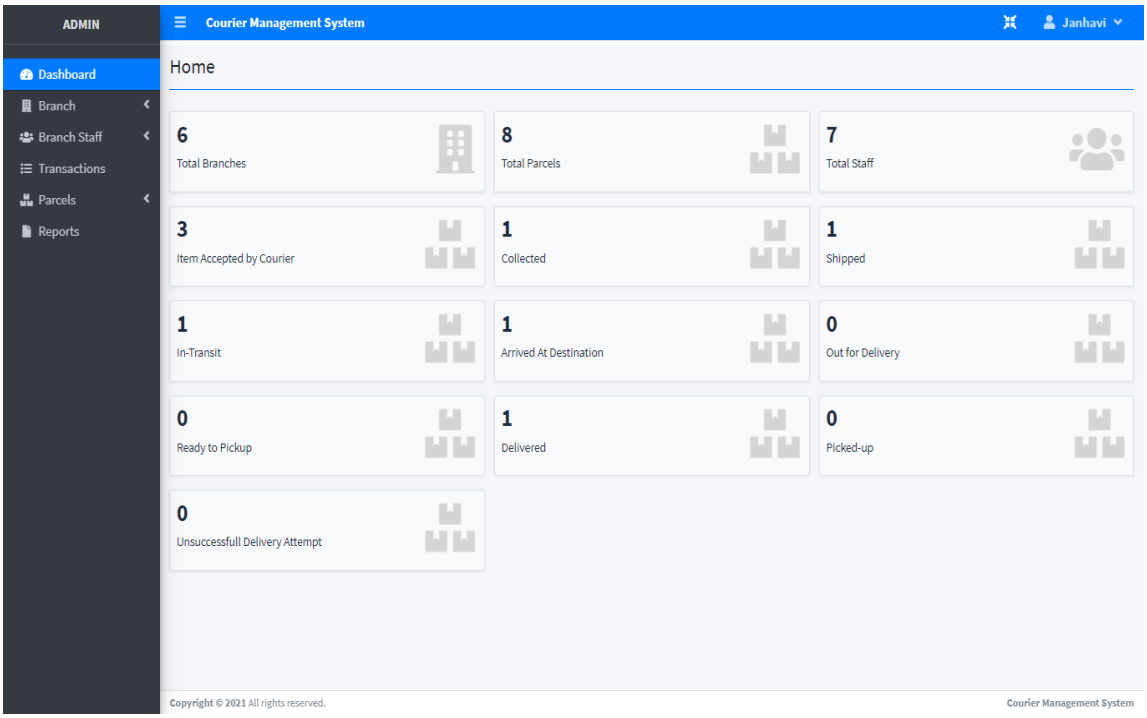
3.Customer(User) Module

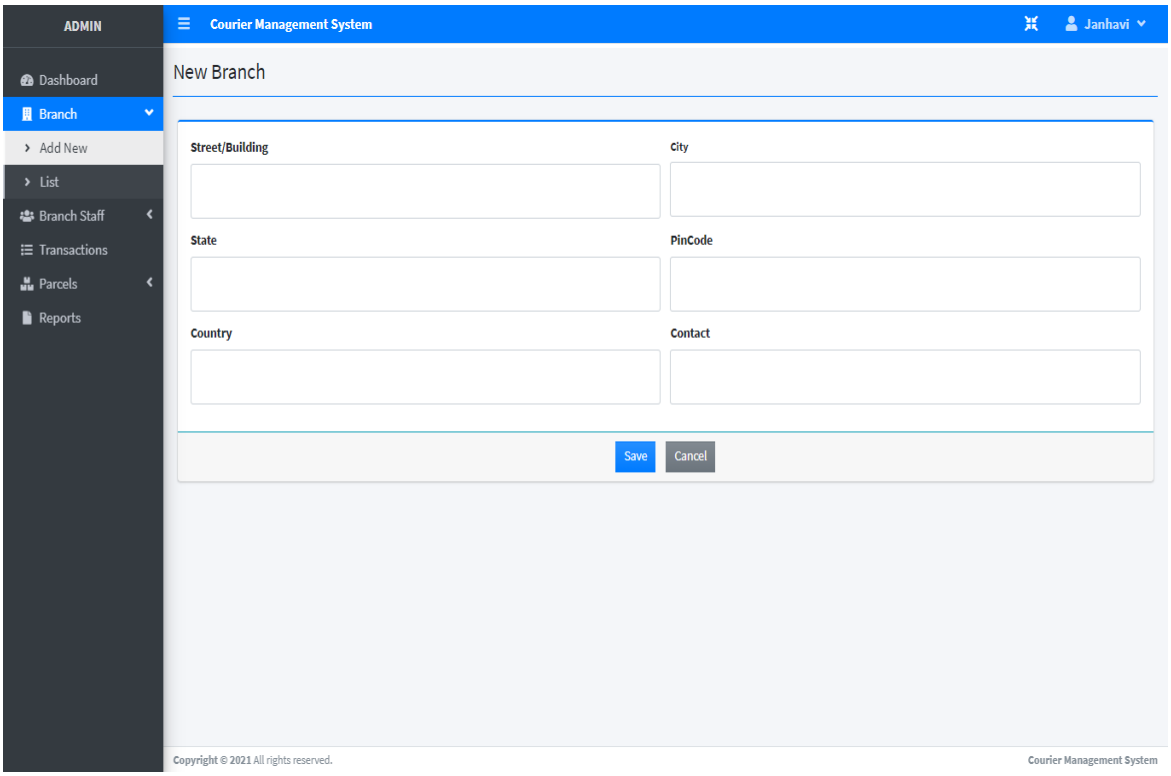
In this module, user can view the current delivery status of his parcel by tracking/reference number and also view the different branches of Courier Company.

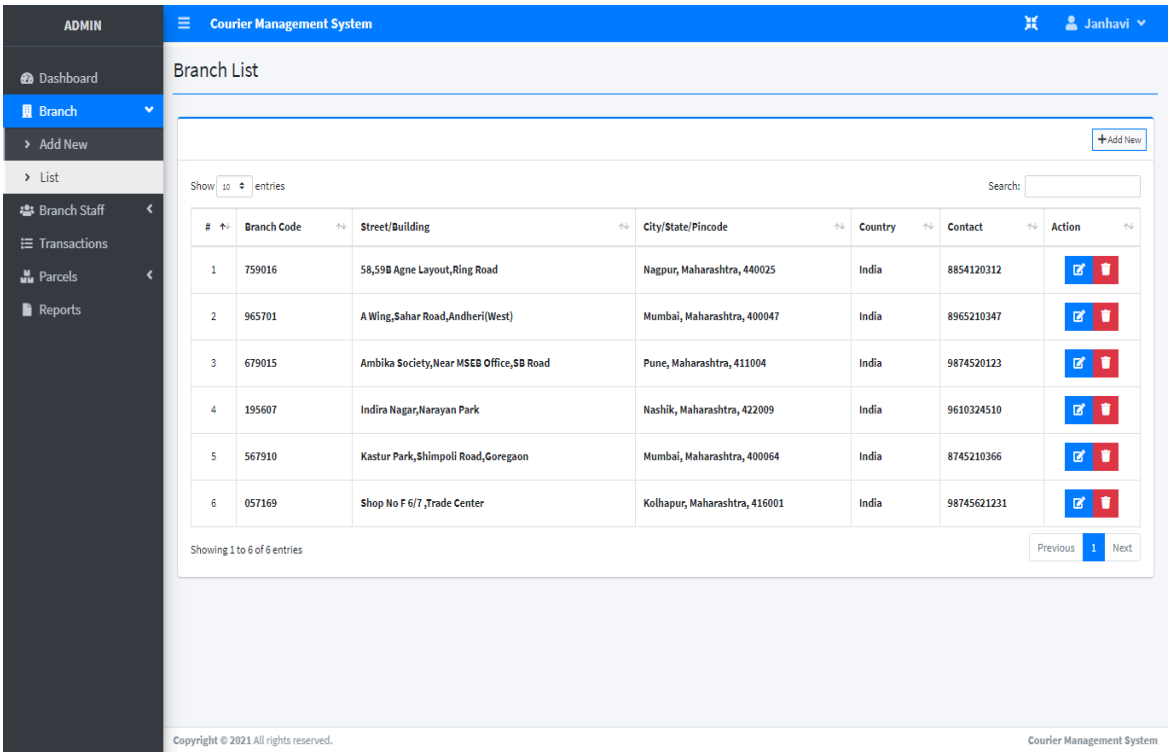
3.13 User Interface Design

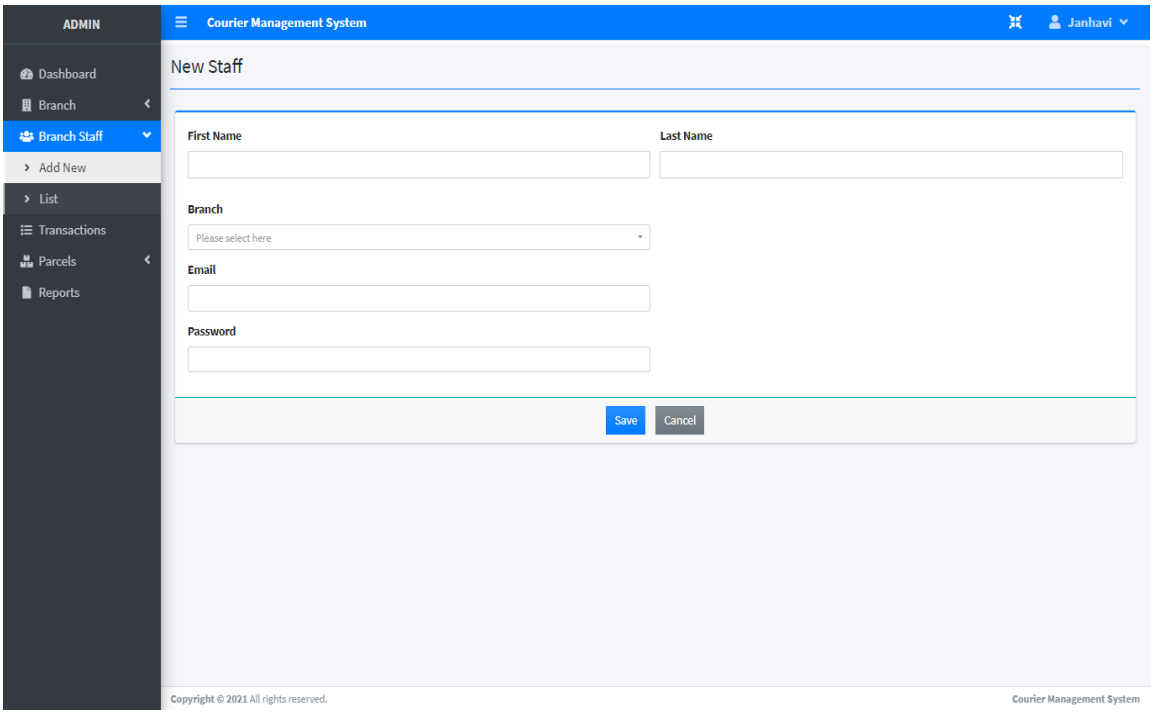


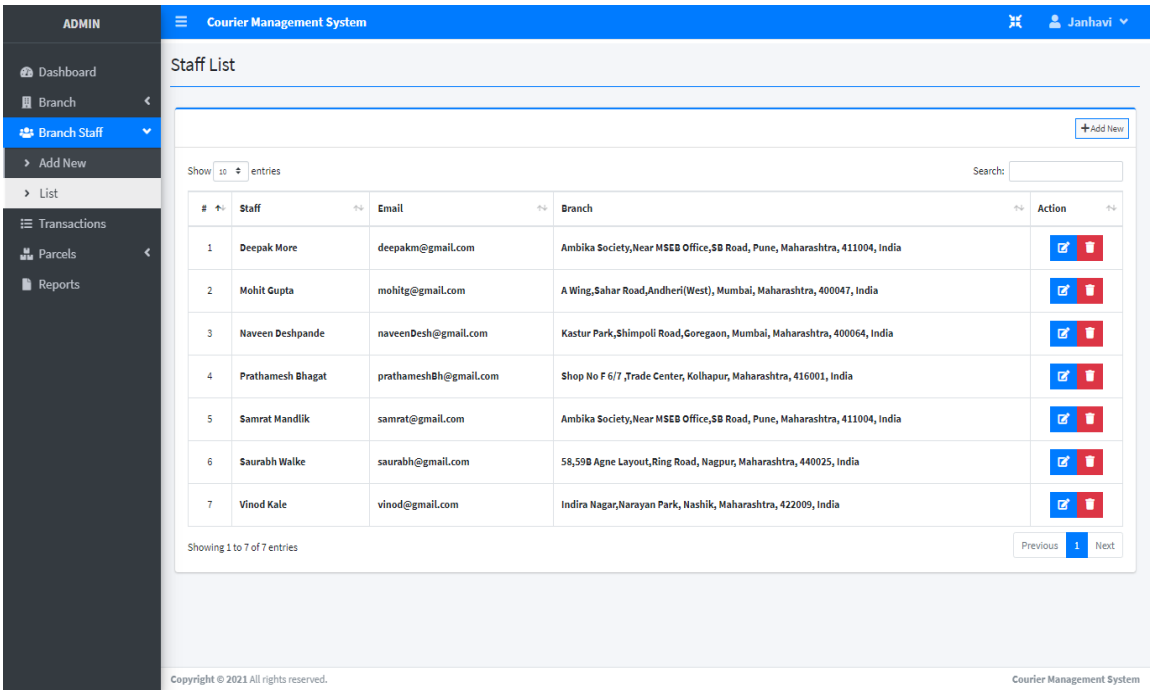

Home | Courier Management Sy
localhost/cms/admin/login.php
Login
Email
Password
Remember Me
Login
Return to homepage

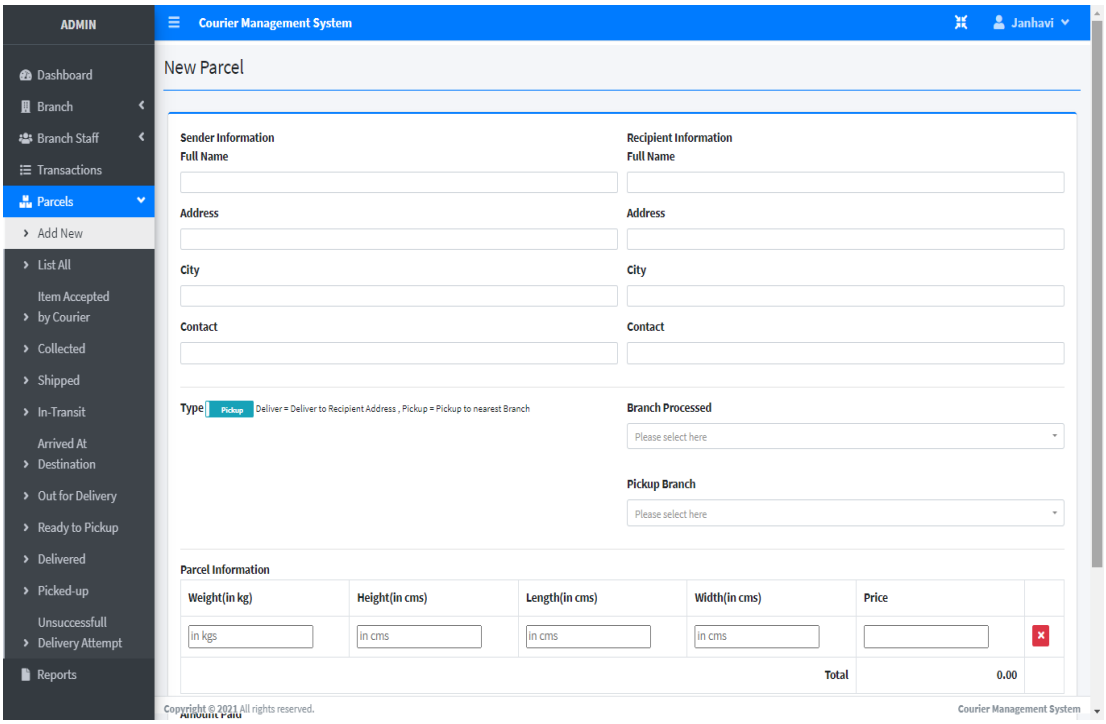

ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Transactions
Parcels
Reports
Home
6
Total Branches
8
Total Parcels
7
Total Staff
3
Item Accepted by Courier
1
Collected
1
Shipped
1
In-Transit
1
Arrived At Destination
0
Out for Delivery
0
Ready to Pickup
1
Delivered
0
Picked-up
0
Unsuccessfull Delivery Attempt
Copyright © 2021 All rights reserved.
Courier Management System

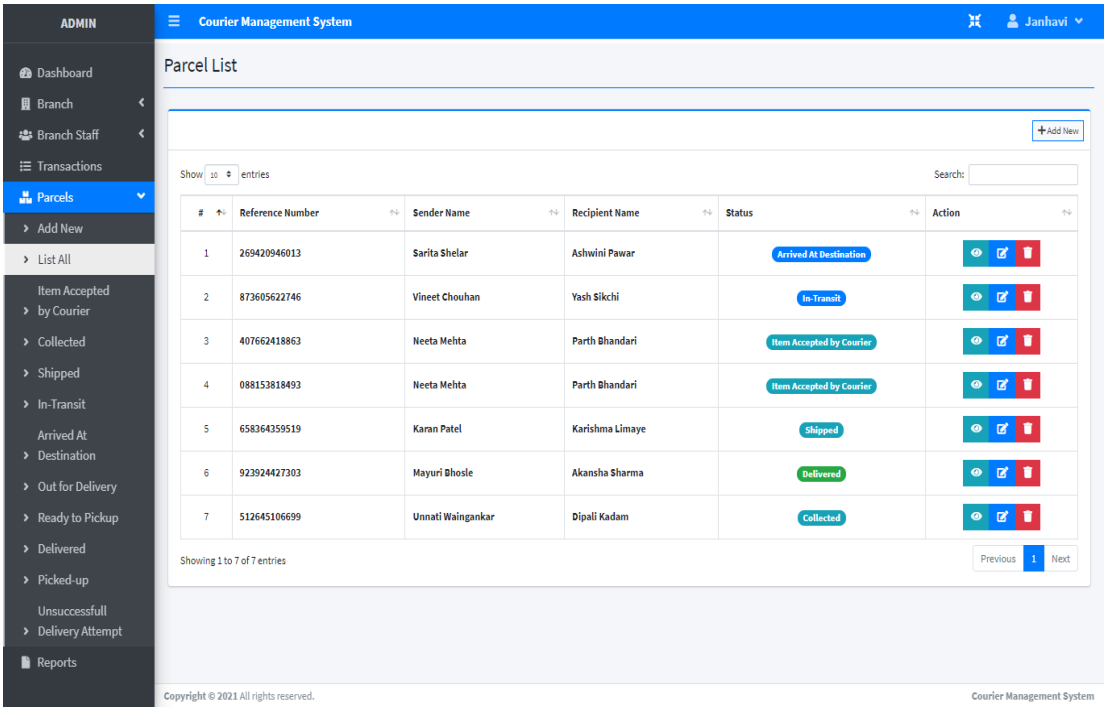

ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Add New
List
Branch Staff
Transactions
Parcels
Reports
New Branch
Street/Building
City
State
PinCode
Country
Contact
Save
Cancel
Copyright © 2021 All rights reserved.
Courier Management System


ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Add New
List
Branch Staff
Transactions
Parcels
Reports
Branch List
+Add New
Show 10 entries
Search:
Branch Code Street/Building City/State/Pincode Country Contact Action
1 759016 58,59B Agne Layout,Ring Road Nagpur, Maharashtra, 440025 India 8854120312
2 965701 A Wing,Sahar Road,Andheri(West) Mumbai, Maharashtra, 400047 India 8965210347
3 679015 Ambika Society,Near MSEB Office,SB Road Pune, Maharashtra, 411004 India 9874520123
4 195607 Indira Nagar,Narayan Park Nashik, Maharashtra, 422009 India 9610324510
5 567910 Kastur Park,Shimpoli Road,Goregaon Mumbai, Maharashtra, 400064 India 8745210366
6 057169 Shop No F 6/7 ,Trade Center Kolhapur, Maharashtra, 416001 India 9874521231
Showing 1 to 6 of 6 entries
Previous 1 Next
Copyright © 2021 All rights reserved.
Courier Management System


ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Add New
List
Transactions
Parcels
Reports
New Staff
First Name
Last Name
Branch
Please select here
Email
Password
Save
Cancel
Copyright © 2021 All rights reserved.
Courier Management System


ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Add New
List
Transactions
Parcels
Reports
Staff List
+Add New
Show 10 entries
Search:
Staff Email Branch Action
1 Deepak More deepakm@gmail.com Ambika Society,Near MSEB Office,SB Road, Pune, Maharashtra, 411004, India
2 Mohit Gupta mohitg@gmail.com A Wing,Sahar Road,Andheri(West), Mumbai, Maharashtra, 400047, India
3 Naveen Deshpande naveenDesh@gmail.com Kastur Park,Shimpoli Road,Goregaon, Mumbai, Maharashtra, 400064, India
4 Prathamesh Bhagat prathameshBh@gmail.com Shop No F 6/7 ,Trade Center, Kolhapur, Maharashtra, 416001, India
5 Samrat Mandlik samrat@gmail.com Ambika Society,Near MSEB Office,SB Road, Pune, Maharashtra, 411004, India
6 Saurabh Walke saurabh@gmail.com 58,59B Agne Layout,Ring Road, Nagpur, Maharashtra, 440025, India
7 Vinod Kale vinod@gmail.com Indira Nagar,Narayan Park, Nashik, Maharashtra, 422009, India
Showing 1 to 7 of 7 entries
Previous 1 Next
Copyright © 2021 All rights reserved.
Courier Management System


ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Transactions
Parcels
Add New
List All
Item Accepted by Courier
Collected
Shipped
In-Transit
Arrived At Destination
Out for Delivery
Ready to Pickup
Delivered
Picked-up
Unsuccessfull Delivery Attempt
Reports
New Parcel
Sender Information
Full Name
Address
City
Contact
Recipient Information
Full Name
Address
City
Contact
Type
Pickup
Deliver = Deliver to Recipient Address , Pickup = Pickup to nearest Branch
Branch Processed
Please select here
Pickup Branch
Please select here
Parcel Information
Weight(in kg)
Height(in cms)
Length(in cms)
Width(in cms)
Price
in kgs
in cms
in cms
in cms
Total
0.00
Copyright © 2021 All rights reserved.
Courier Management System


ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Transactions
Parcels
Add New
List All
Item Accepted by Courier
Collected
Shipped
In-Transit
Arrived At Destination
Out for Delivery
Ready to Pickup
Delivered
Picked-up
Unsuccessfull Delivery Attempt
Reports
Parcel List
+Add New
Show 10 entries
Search:
Reference Number Sender Name Recipient Name Status Action
1 269420946013 Sarita Shelar Ashwini Pawar Arrived At Destination
2 873605622746 Vineet Chouhan Yash Sikchi In-Transit
3 407662418863 Neeta Mehta Parth Bhandari Item Accepted by Courier
4 088153818493 Neeta Mehta Parth Bhandari Item Accepted by Courier
5 658364359519 Karan Patel Karishma Limaye Shipped
6 923924427303 Mayuri Bhosle Akansha Sharma Delivered
7 512645106699 Unnati Waingankar Dipali Kadam Collected
Showing 1 to 7 of 7 entries
Previous 1 Next
Copyright © 2021 All rights reserved.
Courier Management System

3.14 Data Dictionary

Sr no	Column Name	Data Type	Size	Description & Table
1.	Id	Integer	5	Stores Id of users in users table
2.	FirstName	Varchar	40	Stores Name of user in users table
3.	LastName	Varchar	40	Stores lastname of users
4.	Email	Varchar	50	Stores email of users in users table
5.	Password	Varchar	50	Stores password of user in users table
6.	RegDate	Timestamp		Stores reg date of user in users

				table
7.	BranchId	Integer	5	Stores branchId of branch in branch table
8.	BranchCode	Varchar	30	Stores code of branch in branch table
9.	BranchStreet	Varchar	50	Stores sreet of branch in branch table
10.	BranchCity	Varchar	50	Stores city of branch in branch table
11.	BranchState	Varchar	50	Stores state of branch in branch table
12.	BranchPincode	Varchar	30	Stores pincode of branch
13.	BranchCountry	Varchar	50	Stores country

				of branch
14.	BranchContact	Integer	10	Stores contact of branch
15.	DateCreated	Timestamp		Stores created date of branch
16.	CourierId	Integer	5	Stores CourierId of courier in parcel details table
17.	TransactionNumber	Varchar	20	Stores transaction details of courier
18.	SenderName	Varchar	50	Stores sender name in parcel details table
19.	SenderContact	Varchar	10	Stores contact of sender in parcel details table
20.	SenderAddress	Varchar	50	Stores address

				of sender in parcel details table
21.	SenderCity	Varchar	50	Stores city of sender in parcel details table
22.	RecipientName	Varchar	50	Stores name of recipient in parcel details table
23.	RecipientContact	Integer	10	Stores contact of recipient in parcel details table
24.	RecipientAddress	Varchar	50	Stores address of recipient in parcel details table
25.	RecipientCity	Varchar	50	Stores city of

				recipient in parcel details table
26.	Type	Integer	2	Stores in what type the courier will be sent (pickup or delivery)
27.	FromBranchId	Integer	2	Stores name of branch in type of id as foreign key
28.	ToBranchId	Integer	2	Stores name of branch in type of id as foreign key
29.	Amount	Varchar	50	Stores total amount of courier
30.	Balance	Varchar	50	Stores balance amount if any

31.	AmountStatus	Integer	2	Stores the status of amount i.e half or full paid
32.	CourierDate	Timestamp		Stores when courier was booked
33.	Id	Integer	11	Stores tracking id of courier in parcel table
34.	ParcelDetailsId	Integer	11	Stores courierId of courier as foreign key
35.	Reference Number	Varchar	20	Stores reference number or tracking number of courier
36.	Type	Integer	2	Stores in what type the courier will be sent

				(pickup or delivery)
37.	ParcelWeight	Varchar	20	Stores weight of parcel in parcel table
38.	ParcelHeight	Varchar	20	Stores height of parcel in parcel table
39.	ParcelWidth	Varchar	20	Stores width of parcel in parcel table
40.	ParcelLength	Varchar	20	Stores length of parcel in parcel table
41.	Price	Varchar	30	Stores individual price of parcel in parcel table
42.	Status	Varchar	50	Stores status as of courier in

				parcel table
43.	DateCreated	Timestamp		Stores created date of item
44.	ParcelTrackingId	Ineteger	12	Stores tracking id of parcel in parcel tracks table
45.	TrackingId	Integer	10	Stores trackingId of parcel table as foreign key
46.	Status	Integer	04	Stores status of parcel from parcel table
47.	PaymentId	Integer	05	Stores paymentId of parcel in payment table
48.	ParcelDetailsId	Integer	5	Stores id of

				parcel from parcel details table
49.	Amount	Varchar		Stores amount paid by the user in payment table

3.15 Table Specifications

1. Users Table

Column Name	Data type	Size	Constraint
Id	Integer	5	Primary Key
FirstName	Varchar	40	Not Null
LastName	Varchar	30	Not Null
Email	Varchar	50	Not Null
Password	Varchar	50	Not Null
User_type	Integer	1	Not Null
AdminRegDate	Timestamp		

2.Branch Table

Column Name	Data type	Size	Constraint
BranchId	Integer	5	Primary key
BranchCode	Varchar	50	Not Null
BranchStreet	Text	50	Not null
BranchCity	Text	50	Not Null
Branch State	Varchar	50	

BranchPincode	Varchar	50	
BranchCountry	Varchar	50	
Contact	Varchar	10	
DateCreated	Timestamp		

3.Courier Details

Column Name	**Data Type**	**Size**	**Contraint**
Id	Integer	5	Primary key
TransactionNumber	Varchar	20	Not null
SenderName	Varchar	50	
SenderContact	Integer	10	
SenderAddress	Varchar	100	
RecipientName	Varchar	50	
RecipientContact	Integer	10	
RecipientAddress	Varchar	100	
Type	Integer	1	
FromBranchId	Foreign Key	4	

ToBranchId	Foreign Key	4	
Amount	Varchar	4	
Balance	Varchar	4	
AmountStatus	Integer	1	
CourierDate	Timestamp		

4.Parcel Tracking Table:

Column Name	**Data Type**	**Size**	**Constraint**
Id	Integer	11	Primary Key
CourierId	Integer	11	Foreign Key
ReferenceNumber	Varchar	20	
Type	Integer	2	
Weight	Varchar	30	
Height	Varchar	30	
Length	Varchar	30	
Price	Varchar	30	
Status	Integer	2	

DateCreated	Timestamp		

5.PaymentDetails Table:

Column Name	Data Type	Size	Constraint
PaymentId	Integer	5	Primary Key
ParcelDetailsId	Integer	5	Foreign Key
Amount	Varchar	50	
DateCreated	Timestamp		

6.Parcel Tracking Master table

Column Name	Data Type	Size	Constraint
Id	Integer	2	Primary Key
ParcelTrackingId	Integer	5	Foreign Key
Status	Integer	5	
Datecreated	Timestamp		

3.16 Test Procedures and Implementation

Software testing is a critical clement of software quality assurance & represents the ultimate review of specification, design and code generation.

It is the process of executing a program with a primary objective of finding errors. Testing gives the guarantee that the software does not fail and runs according to its specification and in the way the end user expects.

This can be done by various software testing techniques which provide a systematic guidance for designing tests that exercise the internal logic of software components, and exercise the input and output domains of the program to uncover errors in programming

functions, behavior and performance.

Testing is the exposure of system to trial input to see whether it produces correct output. Testing is the process of detecting presence of faults. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to your customer. Our goal is to design a series of test cases that have likelihood of finding errors. That's where Software testing Techniques enter into the picture.

A set of test cases designed to exercise both internal login and external requirements is designed and documented, expected results are defined and actual results are recorded.

Testing Objectives:-

The testing objectives are summarized in the following three steps:

1. Testing is the process of executing a program with the intent of finding a bug.

2. A good case is one that has a high probability of finding an as yet undiscovered error.

3. A successful test is the one that uncover yet an undiscovered error

Unit testing:

Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code usually at the functional level. In an object-oriented environment, this is usually at class-level and the minimal unit tests include the constructors and destructors.These type of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected.

One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is

used to assure that the building blocks of the software work independently of each other.

Integration Testing:

Integration Testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an interactive way or all together ("big bang"). Normally the former is considered a better. practice since it allows interface issues to be localized more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively user groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a software. System Testing: System Testing tests a completely

integrated system to verify that it meets its requirements. The testing phase is an important part of software development, It is the process of finding errors and missing operations and also a complete verification to

determine whether the objectives are met and the user requirements are satisfied.

Acceptance Testing:

Acceptance testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of the program is not emphasized. Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Acceptance testing is performed along with the client to show that to see that all requirements are satisfied whatever may be the attributes its working well provided all the attributes are valid. If not it displays corresponding messages for getting valid attributes.

Alpha Testing:

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at developers site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing. Beta Testing: Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the products have few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback filled to a maximal number of future users.

Usability Testing:

Usability testing is needed to check if the user interface is easy to use and understand. It is connected mainly with the use of the application.

Security Testing:

Security testing is essential for software that processes confidential data to prevent system intrusion by hackers.

White Box Testing :

This is the unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. We tested stepwise every piece of code, taking care that every statement in the code is executed at least once; the white box testing is also called glass box Testing.

Black Box Testing :

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details as statement level. Output for a given set of input combinations are forwarded other module.

TEST CASES

Test Case Id	Scenario To test	Steps To Perform	Expected Result	Actual Result	Status
1	Show tracking status on homepage	Loading tracking status on home screen	Displayed Status	Data Displayed	Pass
2	Log-in into application as Employee/Manager/Admin	1.Open the log in page of the application. 2.Enter the valid user name. 3.Enter valid password. 4.Click on	Application should except valid user name and valid password entered by user and should redirect user	Application should except valid user name and valid password entered by user and should redirect user	Pass

		Log in button	to respected dashboard	to respected dashboard	
3	Log-in into application	1.Open the log in page of the application. 2.Enter in-valid user name. 3.Enter valid password. 4.Click on Log in button.	Application should not accept invalid user name. Application should throw message “Invalid Credentials	Log in denied with appropriate message	Pass
4	Log in into application	1.Open the log in page of the application.	Application should not accept invalid user	Log in denied with appropriate message	Pass

		2.Enter the valid user name. 3.Enter in-valid password. 4.Click on Log button	name. Application should throw message “Invalid Credentials”.		
5	Dashboard	Check everthing is working on dashboard	Dashboard should work fine	Dashboard working as expected	Pass
6	Check whether all status of courier is displayed with count	Display courier status	Status should be displayed in right manner	Status displayed	Pass
7	If user is admin Add branch details to the dashboard page	Check whether branch	Branch details added with	Data Added	Pass

		details are added according to validation	validation		
8	If user is admin Display branch details in list on click of branch list	Check all branches are displayed	Should display all added branches	Displaying all added branches	Pass
9	If user is admin add employee details in respective branch	Check whether employee are added according to branch	Employee should be added	Employee is added sucessfully	Pass
10	If user is admin display employee list	Check whether all employee are added to list	Employee should be displayed in list	Employee list is being displayed	Pass
11	Add courier or parcel	Check	Parcel	Parcel is	Pass

		whether the parcel is getting added along with its properties	should be added	getting added	
12	Check whether added parcel is displayed in list of parcels	Added parcel should be displayed in list of parcels	Display list of parcels	List of parcels getting displayed	Pass
13	If user is employee then display only parcels related to his branch	Check whether branch related parcels being displayed	Branch related parcels should be displayed	Branch related parcels getting displayed	Pass
14	Get all courier related tracking details seperately	Check whether courier is	Courier should be displayed	Courier is getting displayed	Pass

		being displayed according to status	according to their respective status	according to status	
15	Get tracking number for each courier	Check whether unique tracking number is generated for each courier	Display unique tracking number	Unique tracking number being displayed	Pass
16	Transaction details	Check whether transactions related to courier is stored and getting displayed	Transaction details should display correctly	Transactions getting displayed	Pass

17	Reports	Viewing reports	Reports showing correctly	Reports working as expected	pass

CHAPTER 4

USER MANUAL

4.1 User Manual

For any system to be successful it is important that the intended user find the system easy to operate. The purpose of the user manual is to make user acquainted with the system and help user understand the system and operate it conveniently. The User Manual is prepared reflexively because it is an item that must accompany every system.

The manual contain several screenshots that describes how to use the entire system. This Manual helps user to navigate efficiently through the system and help user to solve issues wherever they occur.

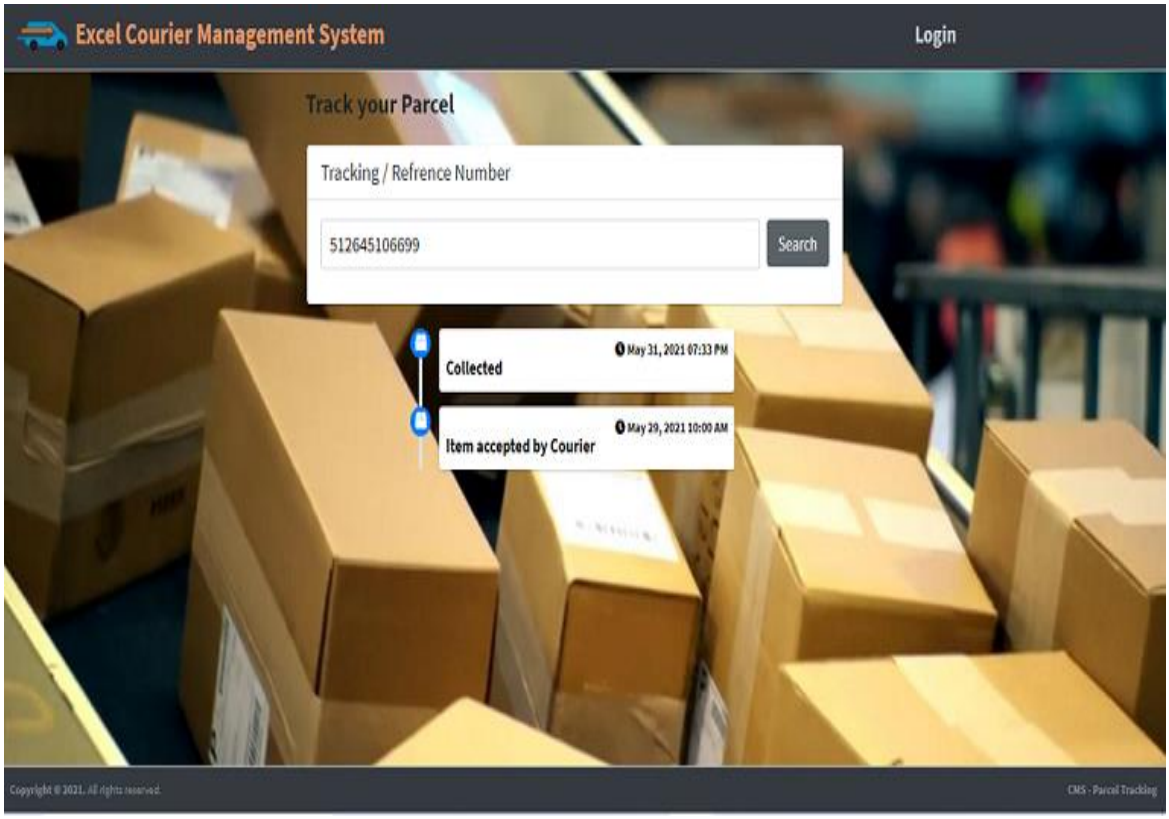
The system contains following users:

1) Admin

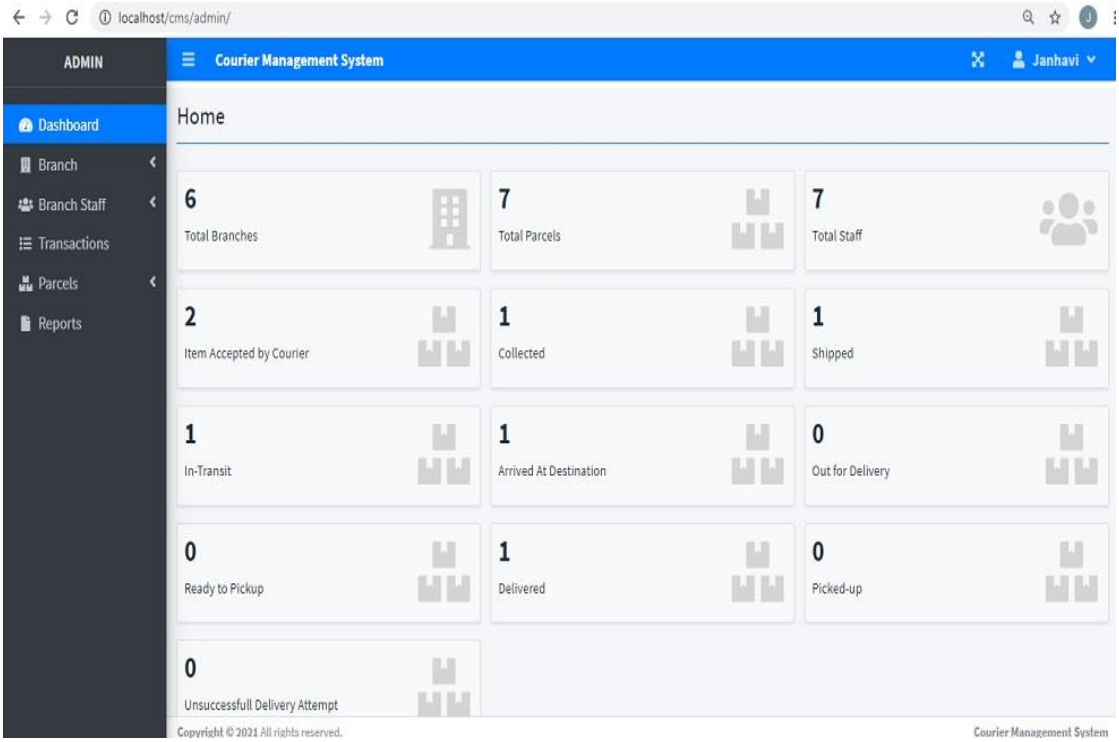
2) Customer

3) Employee

Enter tracking/reference number generated while booking courier.You will get courier details.

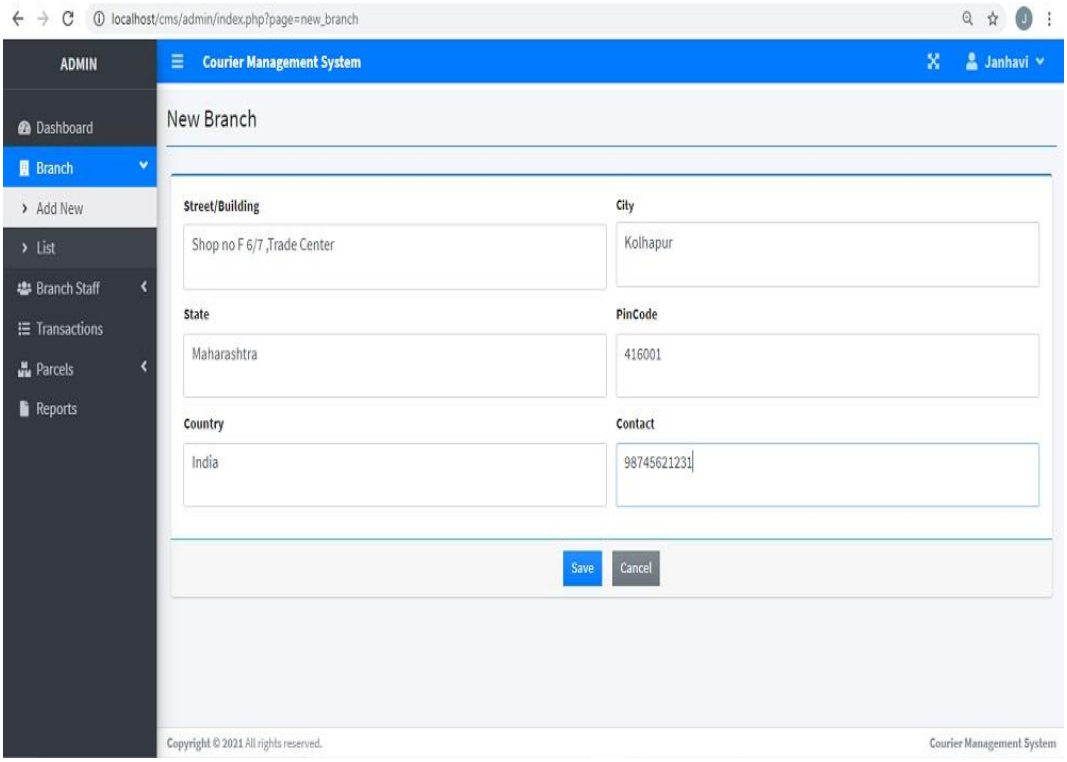


Log in into the system using User credentials. You are redirected to dashboard after successful credential verification of Username and Password.



How to add and view branch:

1.Select Add branch from the dashboard and you will get a form.

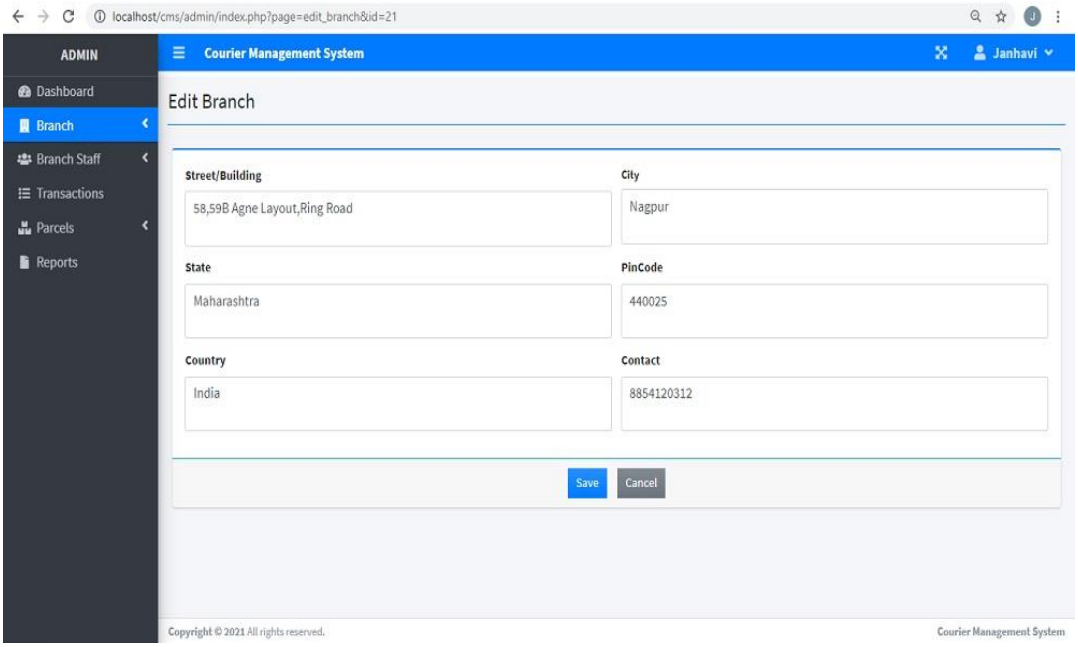


2.Add data and Save. Message of data successfully added generated.

Click Now on the branch list

3.To update a branch:

Click on edit icon to the right of the branch displayed and data will be updated.

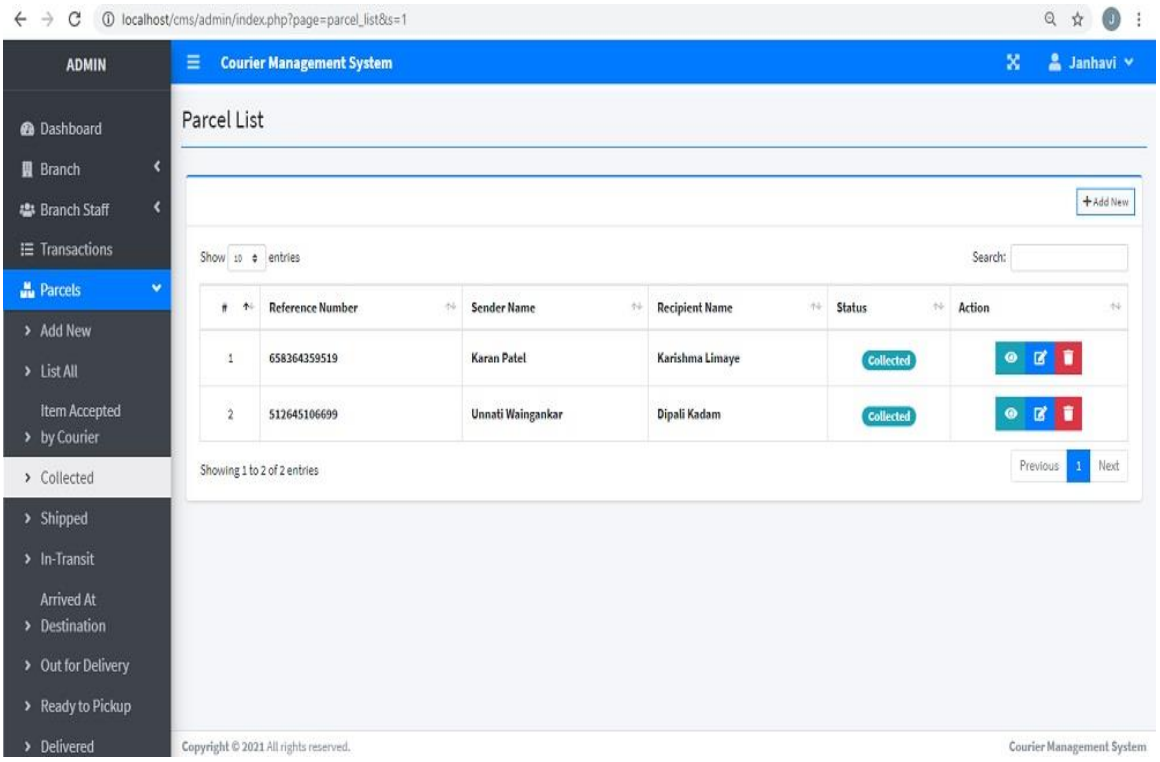


4.How to add and view a courier:

- Select Add courier from the dashboard and you will get a form.
- Add data and Save. Message of data successfully adde generated. Click Now on the branch list

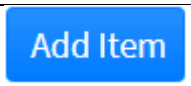
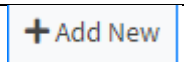
5.To view courier status:

1. Click on any courier statuses mentioned under parcel details you will get such scrren for displaying status



4.2 Operations Manual

There are various symbols and buttons on the web application. Here are their descriptions :

Button	Use
	To view details of parcel,branch etc
	To edit details of parcel,staff etc
	To delete any item
	To add multiple couriers
	To save any type of data
	To add new parcel,staff,branch etc

4.3 Program Specifications

1) Add User Details

Module	Admin
Program Name	Add User Details
Purpose	Add user details including employee,customer etc
Input Details	The required fields should not be blank and the user should provide valid data for each field.
Output	Details of user are stored in respective users table

2)Add Branch Details

Module	Admin
Program Name	Add Branch Details
Purpose	Add all branch related details
Input Details	The required fields should not be blank and the admin should provide valid data for each field

Output	Details of branch stored in branch table

3)Add Courier Details

Module	Admin/Employee
Program Name	Add all courier related details with details of sender and recipient
Purpose	To store all courier related details
Input Details	The required fields should not be blank and the user should provide valid data for each field
Output	Details of courier stored in courier table

4)Edit Details

Module	Edit Master data from tables
Program Name	Edit Masters
Purpose	Add, Edit or Delete master data from the master tables.
Input Details	The admin head should specify the modifications to be made in the master data.
Output	The data from the master table(s) is updated

5)Update status

Module	Admin or Employee
Program Name	Update status of courier
Purpose	Store status of courier in respective table
Input Details	Input accurate courier status
Output	Status updated in parcel tracks table

6)Report generation

Module	Reports
Program Name	Report Generation
Purpose	Show reports of couriers based on status selected

Input Details	Data will be fetched from the databse based on the category selected
Output	Reports will be displayed in

Drawbacks and Limitations

1.Forgot Password functionality is not available.User needs to go to the database to change password.

2.Calculation of price is not done automatically.User needs to enter price manually according to weight parameters of courier.

3.Customer can view only tracking details which involves only status and date of courier.

4.Customer login functionality is not available.

5.Limited generation of reports.

Proposed Enhancements

Following are future changes that are suggested:

1)Customer login should be available.He should be able to book his own courier in system

2)Forgot password functionality should be added.

3)When courier related parameters are entered like width,height,length price should be calculated automatically.

Conclusions

Working on the project was a good experience. I understand the importance of planning and designing as part of software development. During the project, the real importance for the following principle of System analysis and design drawned on me.

Some brief points about the project:

- Cms is a system which allows to track courier efficiently along with displaying dates of status
- The system has a part payment functionality which allows user to make part payment.
- The system has barcode which displays courier reference no which is unique for each parcel.
- The system reduces time of courier booking and tracking process.

Bibliography:

The completion of this project would not have been possible without the following sources listed below:

1. www.google.co.in

2. www.w3school.com

3. www.colorlib.com

4. www.htmlgoodies.com

5. www.getbootstrap.com

6. www.codeofaninja.com

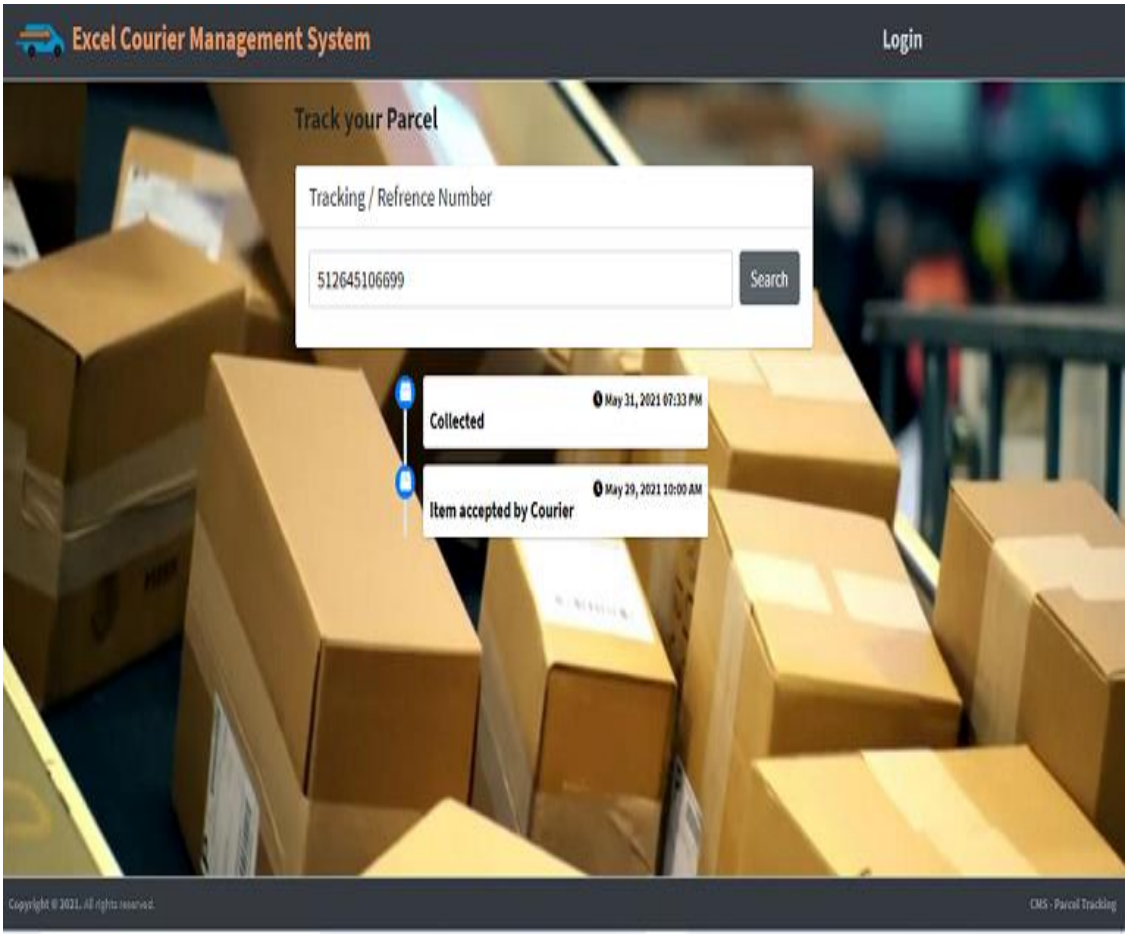
7. www.freecodecamp.org

ANNEXURES

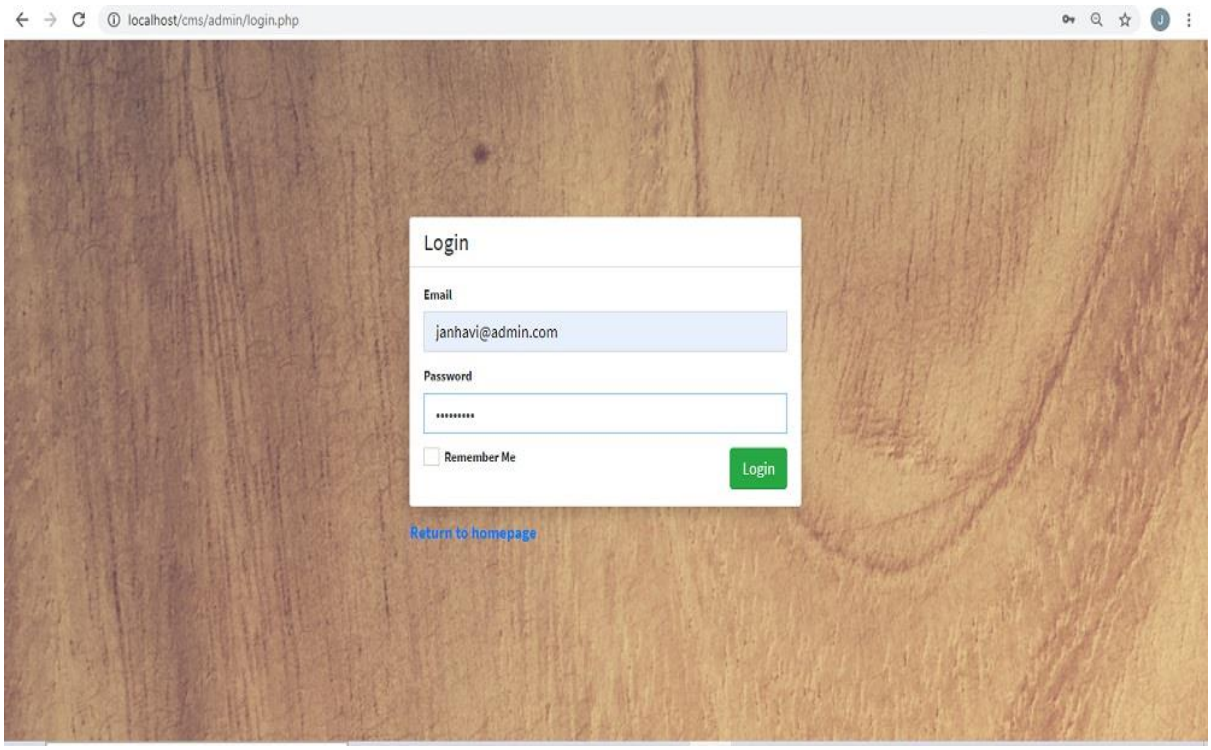
ANNEXURE 1

USER INTERFACE SCREEN

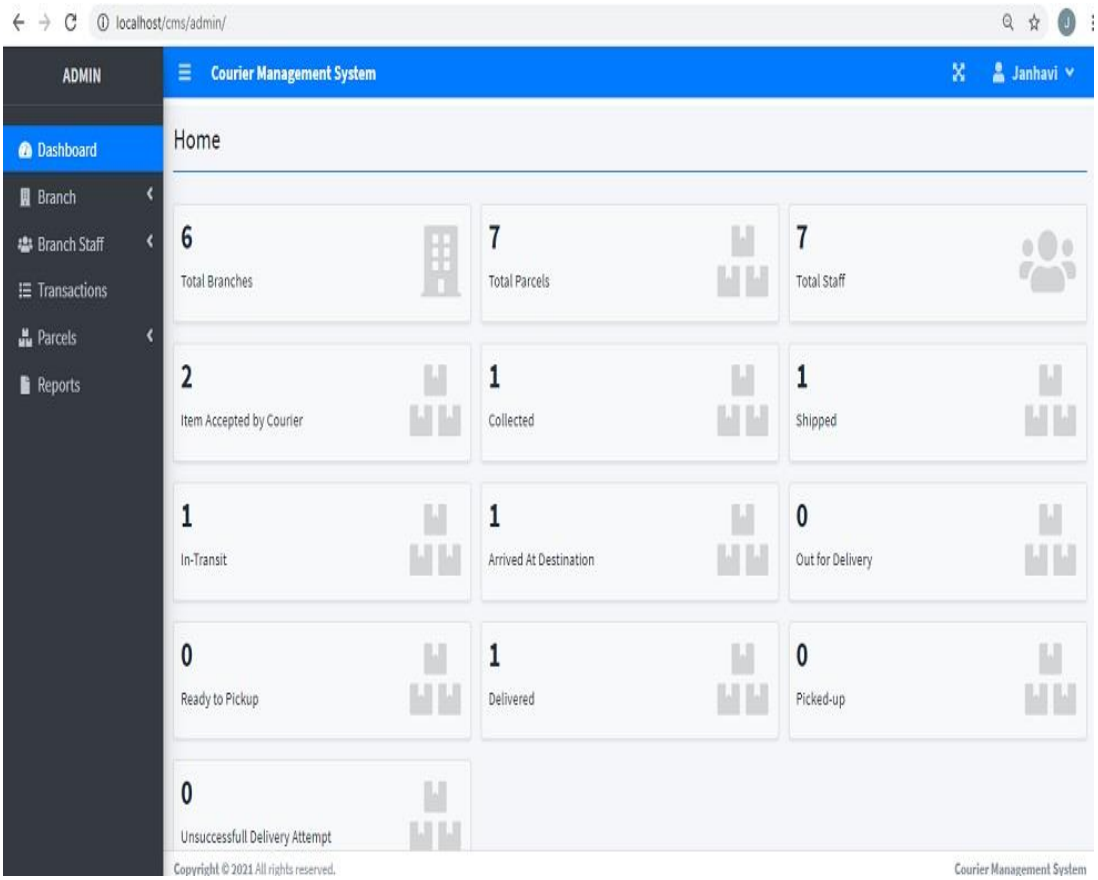
Homepage


Excel Courier Management System
Login
Track your Parcel
Tracking / Refrence Number
51264510669 9
Search
May 31, 2021 07:33 PM
Collected
May 29, 2021 10:00 AM
Item accepted by Courier
Copyright © 2021. All rights reserved.
CMS - Parcel Tracking

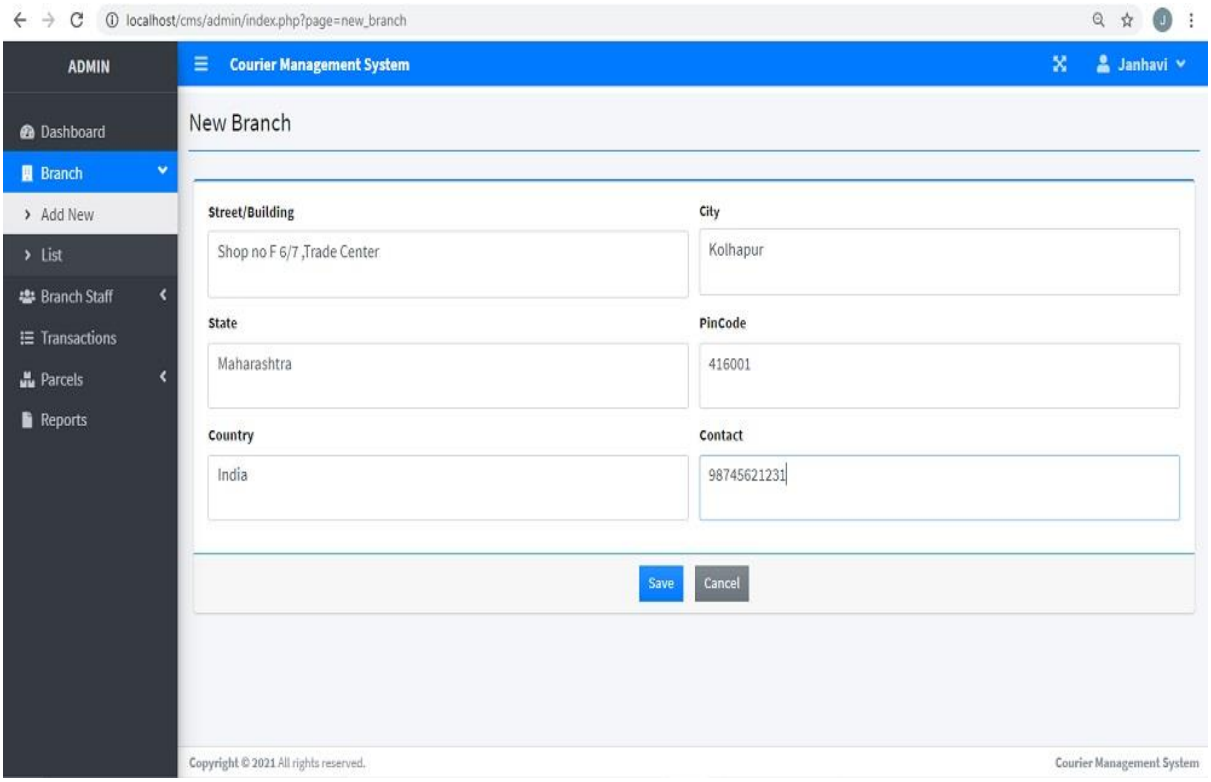
Login Page


localhost/cms/admin/login.php
Login
Email
janhavi@admin.com
Password
Remember Me
Login
Return to homepage

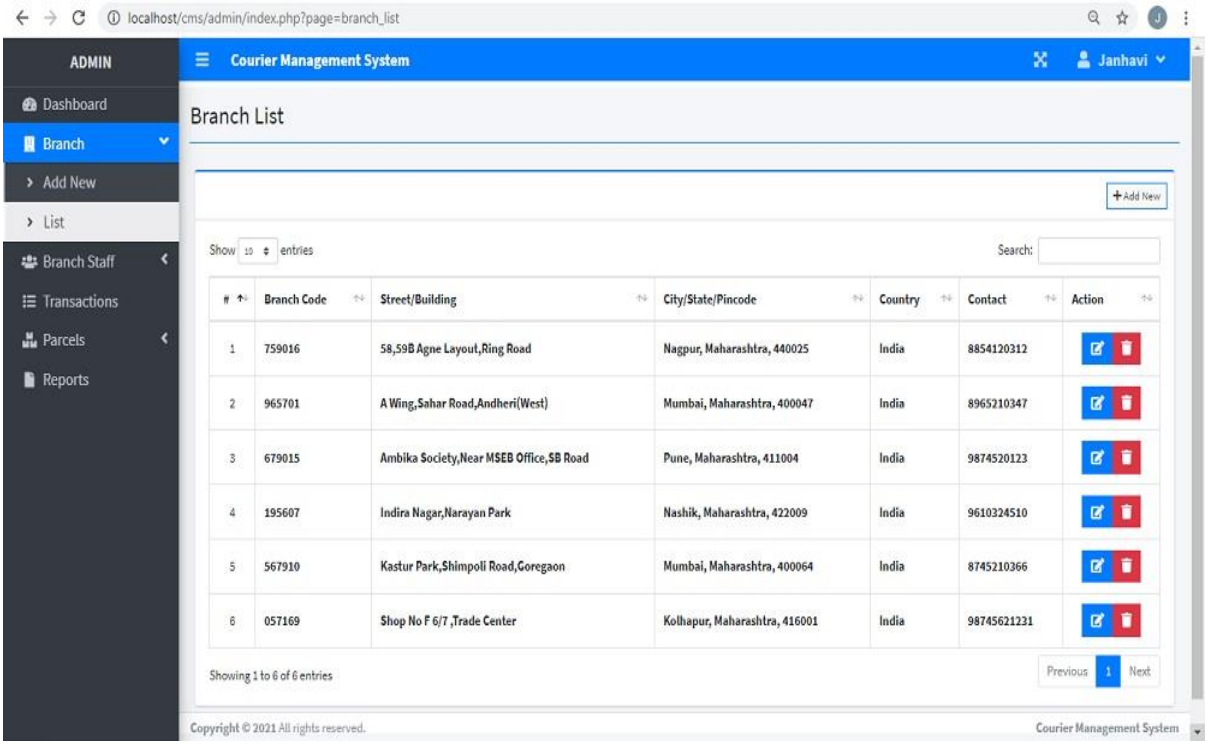
Dashboard


localhost/cms/admin/
ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Transactions
Parcels
Reports
Home
6
Total Branches
7
Total Parcels
7
Total Staff
2
Item Accepted by Courier
1
Collected
1
Shipped
1
In-Transit
1
Arrived At Destination
0
Out for Delivery
0
Ready to Pickup
1
Delivered
0
Picked-up
0
Unsuccessfull Delivery Attempt
Copyright © 2021 All rights reserved.
Courier Management System

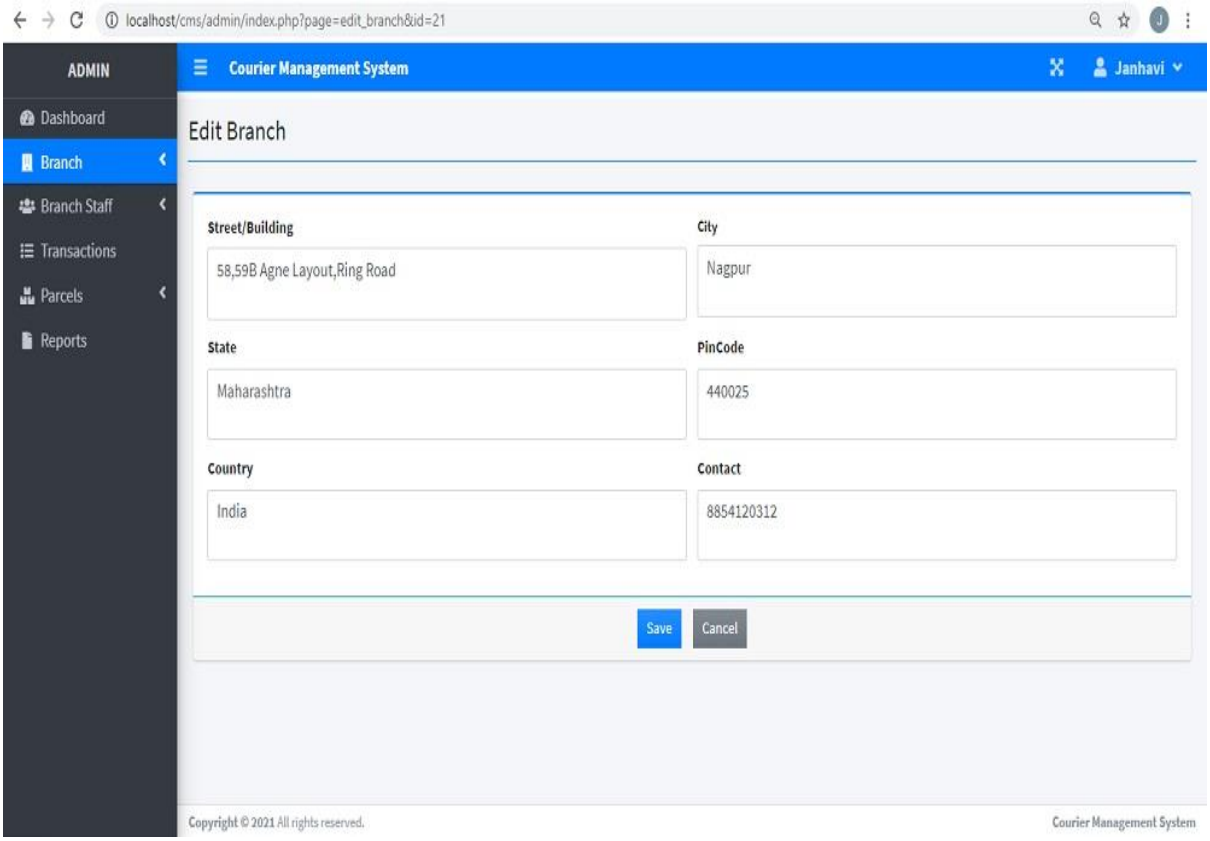
Add branch


localhost/cms/admin/index.php?page=new_branch
ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Add New
List
Branch Staff
Transactions
Parcels
Reports
New Branch
Street/Building
Shop no F 6/7 ,Trade Center
City
Kolhapur
State
Maharashtra
PinCode
416001
Country
India
Contact
98745621231
Save
Cancel
Copyright © 2021 All rights reserved.
Courier Management System

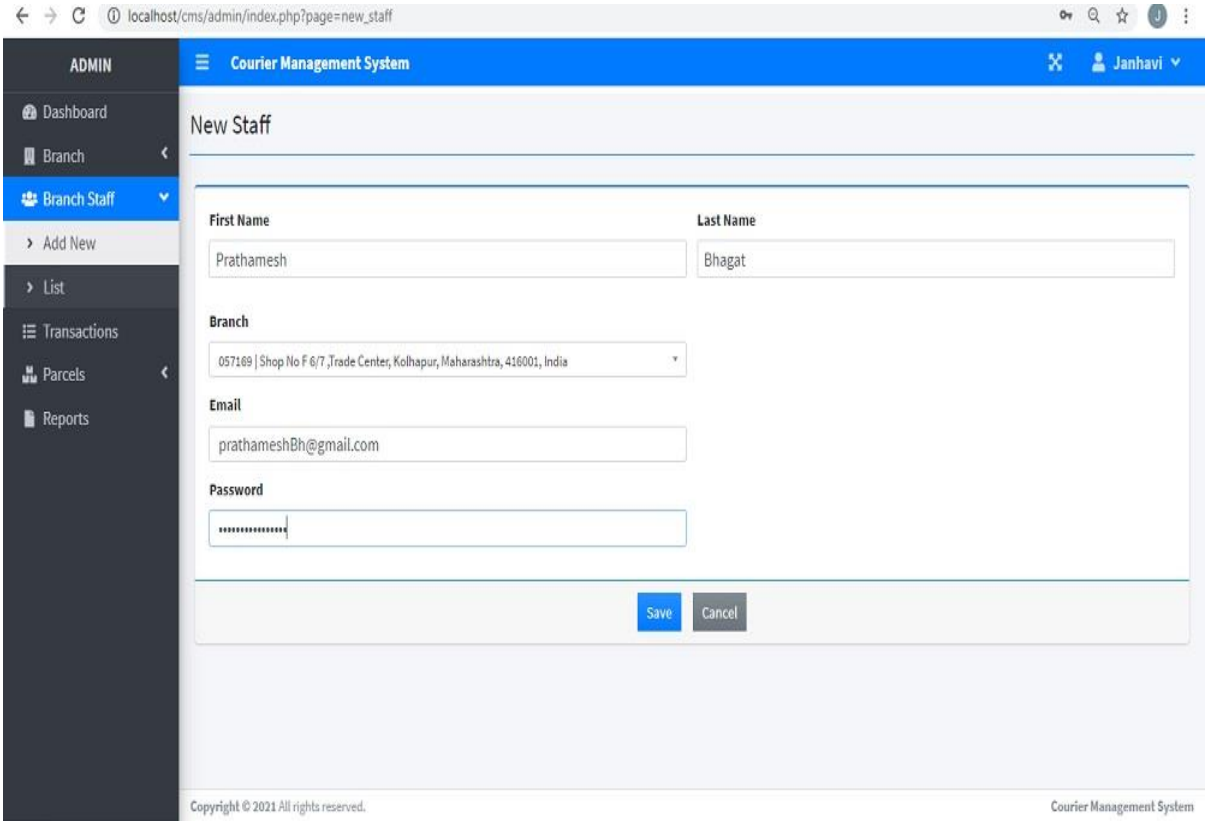
Branch List


localhost/cms/admin/index.php?page=branch_list
ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Add New
List
Branch Staff
Transactions
Parcels
Reports
Branch List
+Add New
Show 10 entries
Search:
Branch Code Street/Building City/State/Pincode Country Contact Action
1 759016 58,59B Agne Layout,Ring Road Nagpur, Maharashtra, 440025 India 8854120312
2 965701 A Wing,Sahar Road,Andheri(West) Mumbai, Maharashtra, 400047 India 8965210347
3 679015 Ambika Society,Near MSEB Office,SB Road Pune, Maharashtra, 411004 India 9874520123
4 195607 Indira Nagar,Narayan Park Nashik, Maharashtra, 422009 India 9610324510
5 567910 Kastur Park,Shimpoli Road,Goregaon Mumbai, Maharashtra, 400064 India 8745210366
6 057169 Shop No F 6/7 ,Trade Center Kolhapur, Maharashtra, 416001 India 98745621231
Showing 1 to 6 of 6 entries
Previous 1 Next
Copyright © 2021 All rights reserved.
Courier Management System

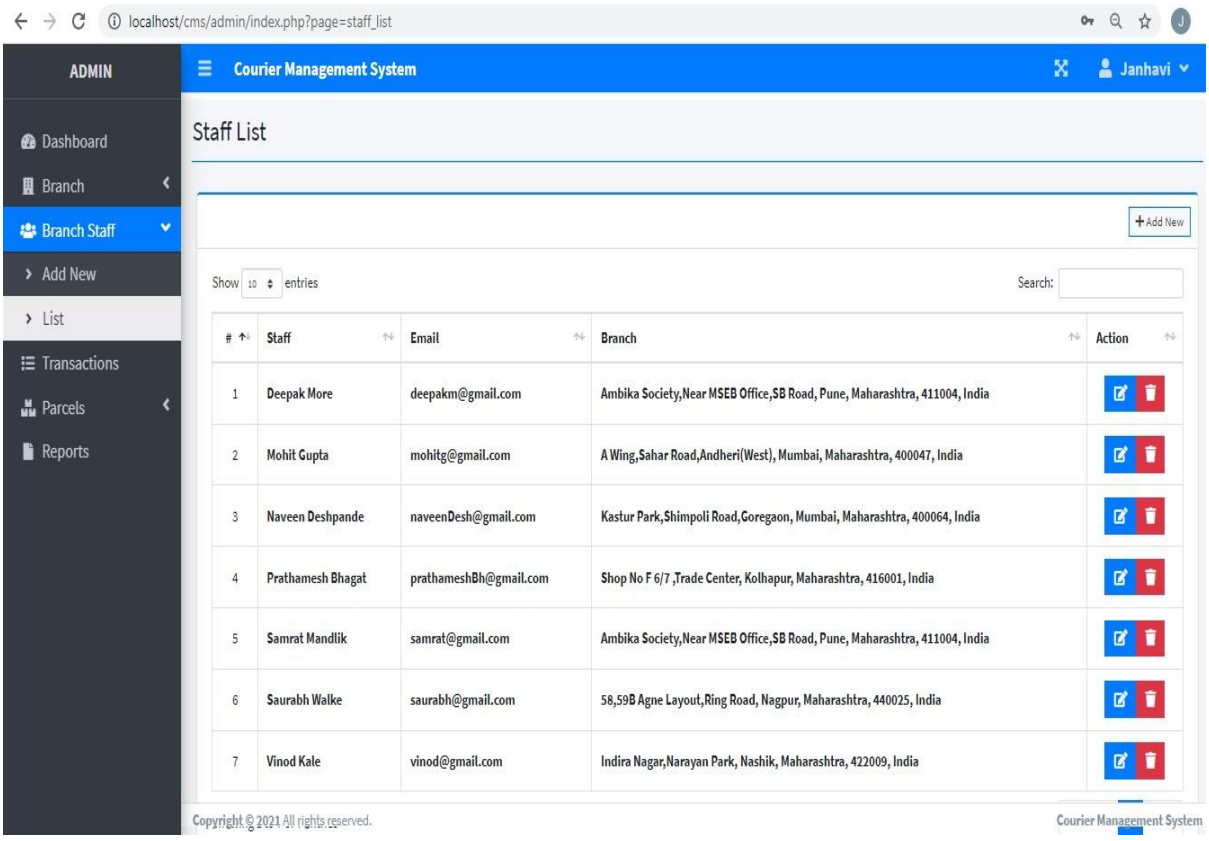
Update branch details



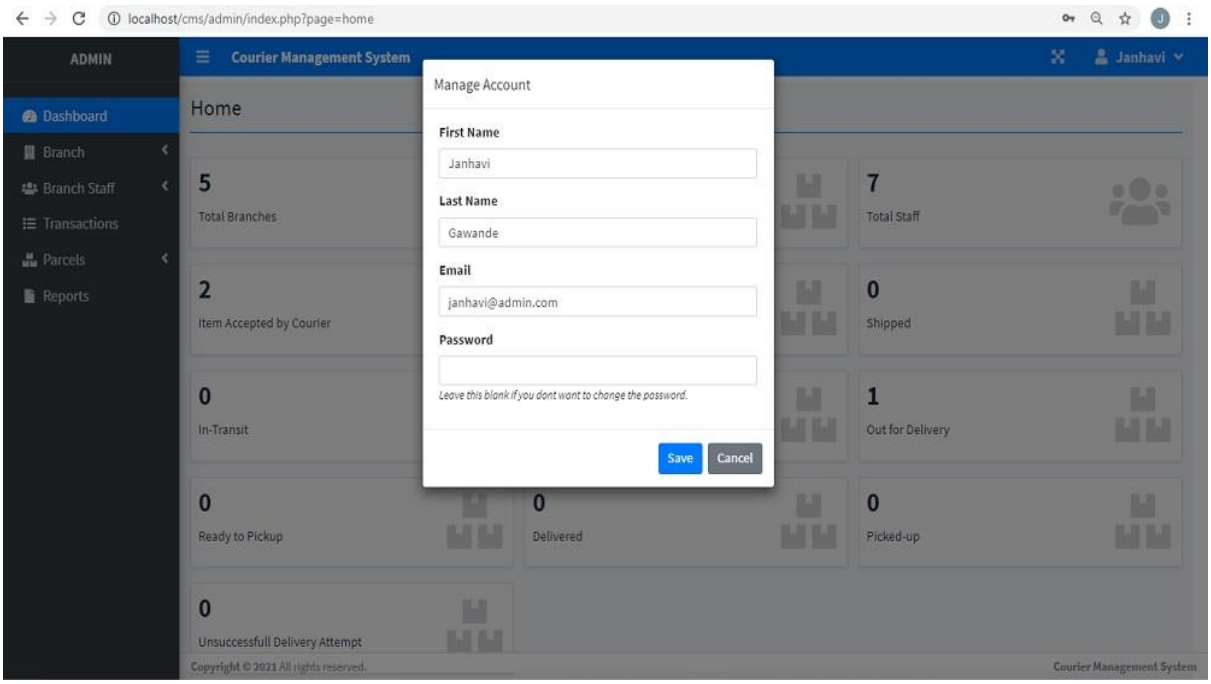
Add Branch Staff



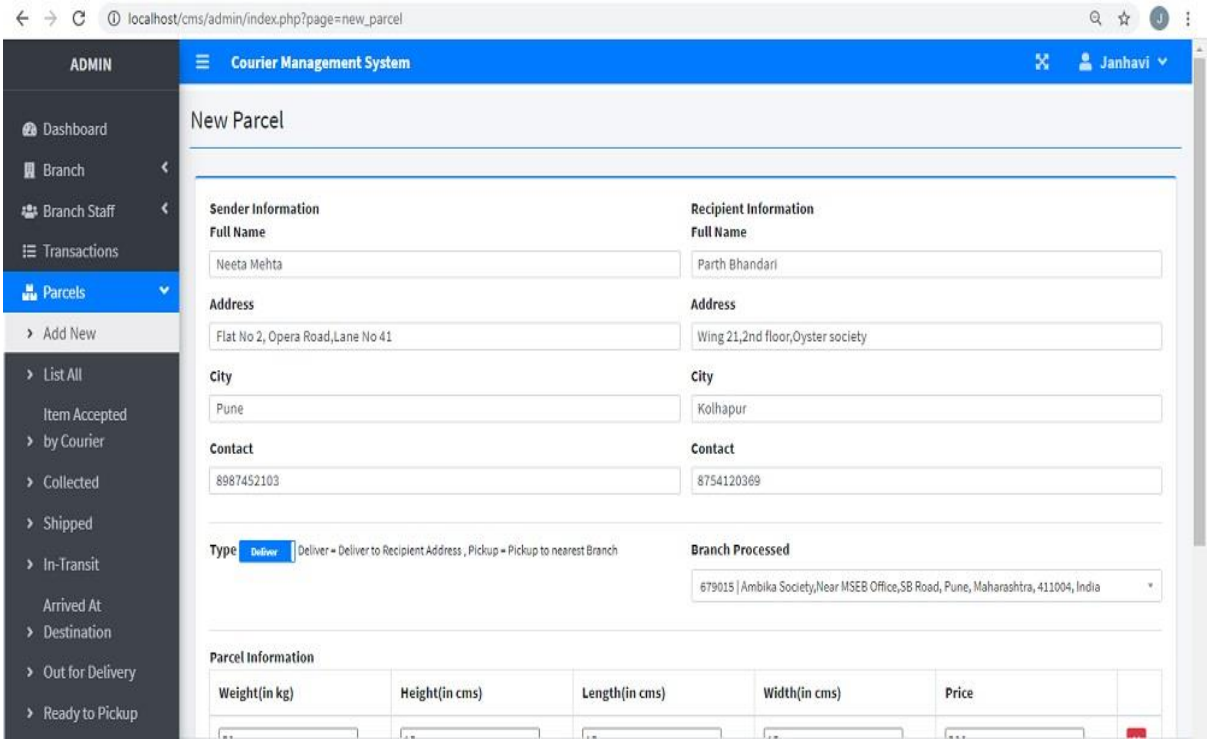
List Branch Staff



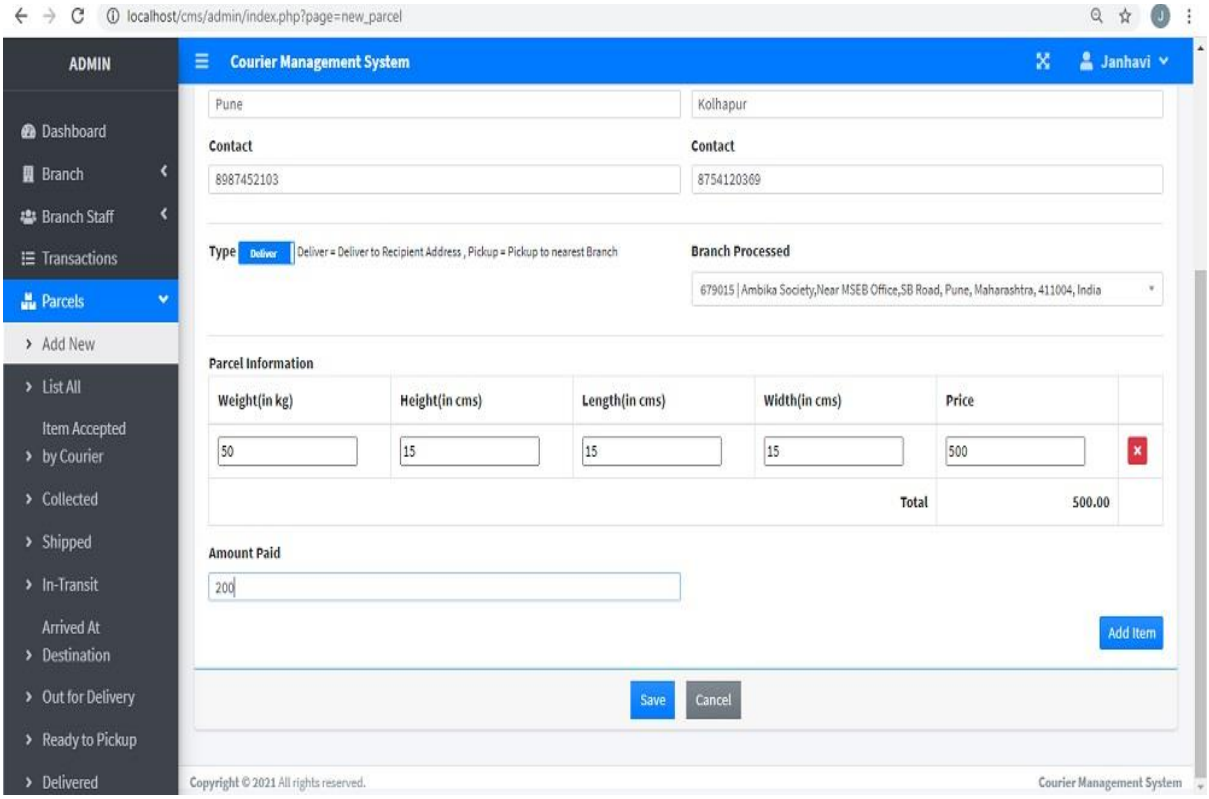
Update admin



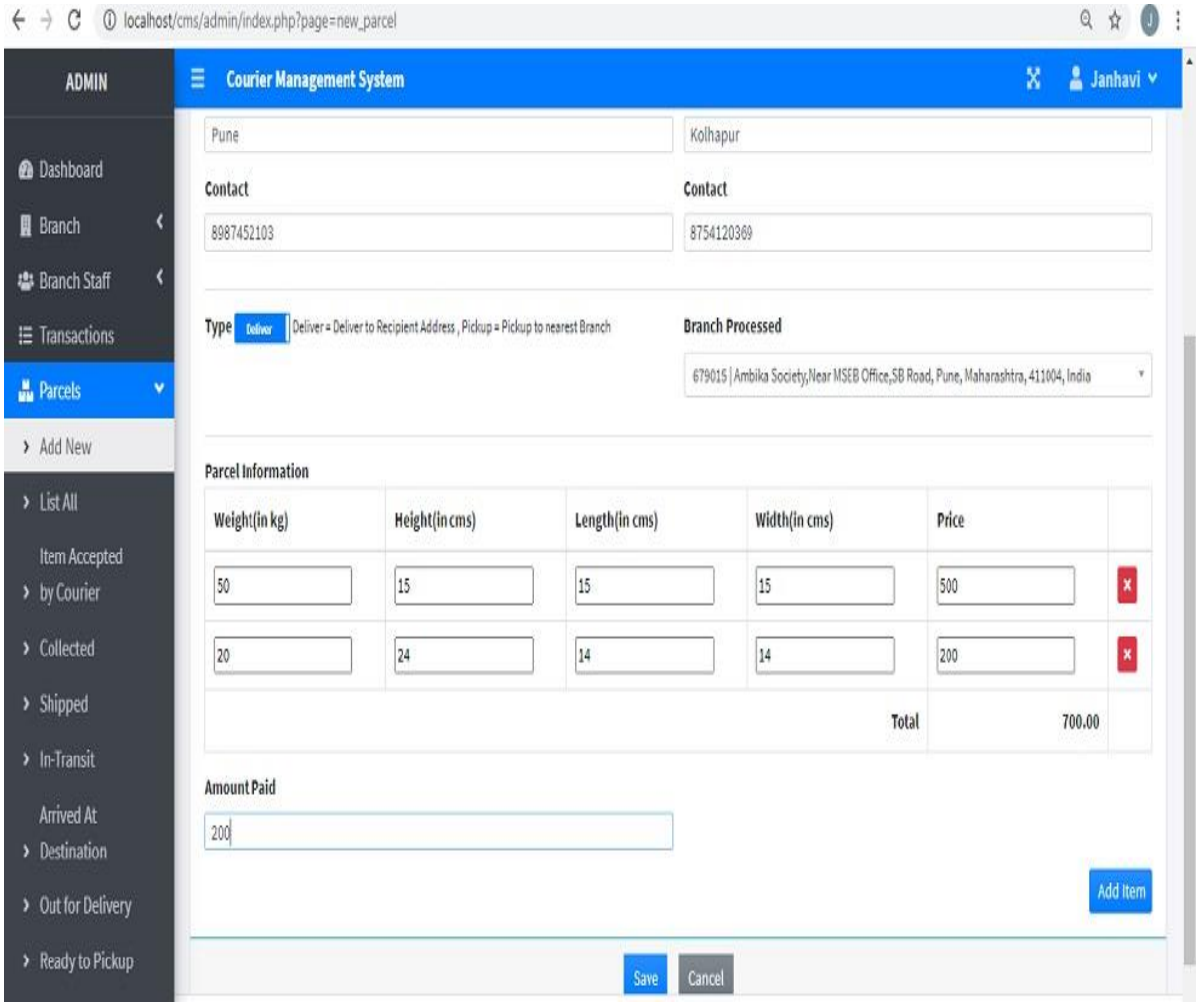
Add new parcel



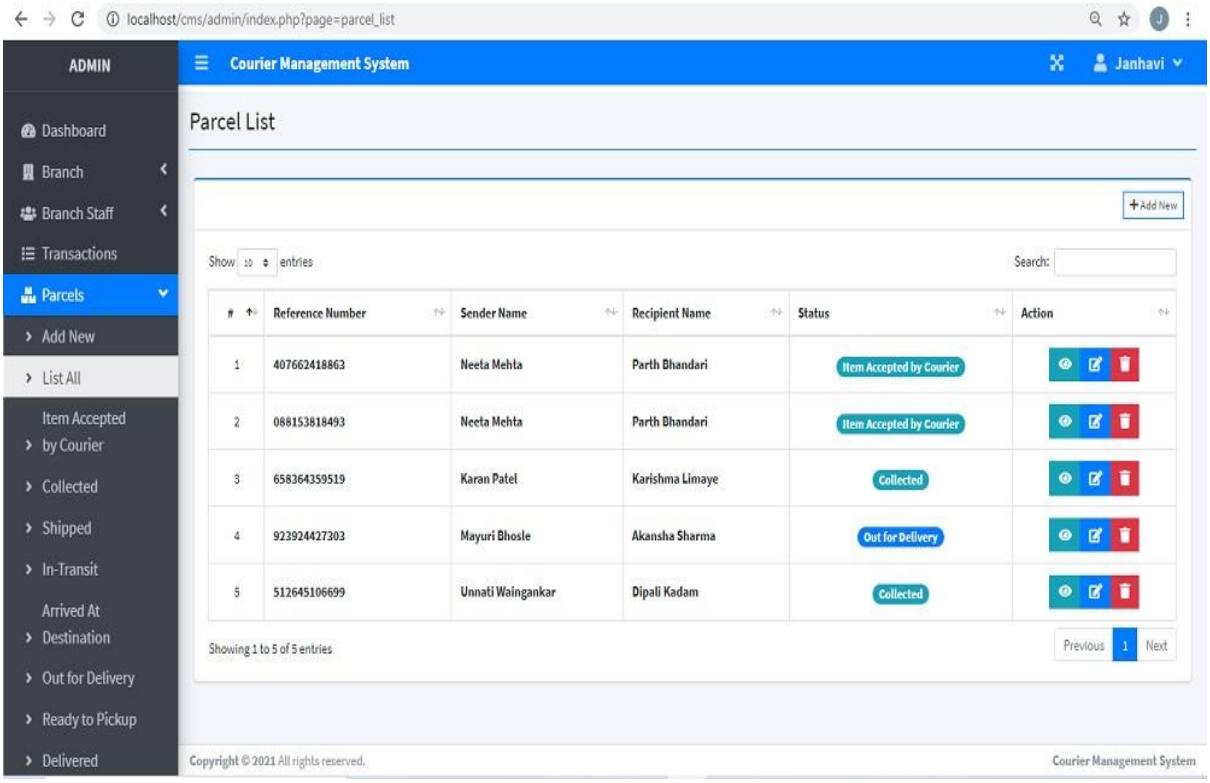
Add new parcel



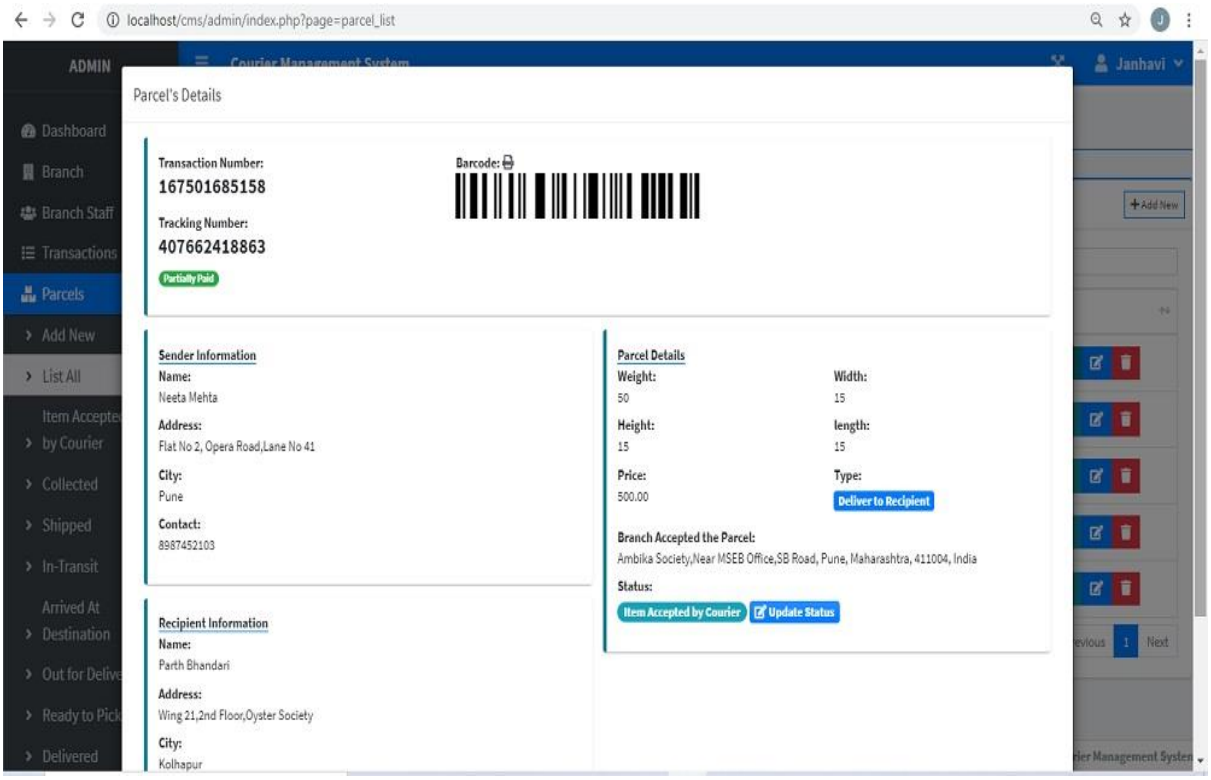
Add multiple parcels



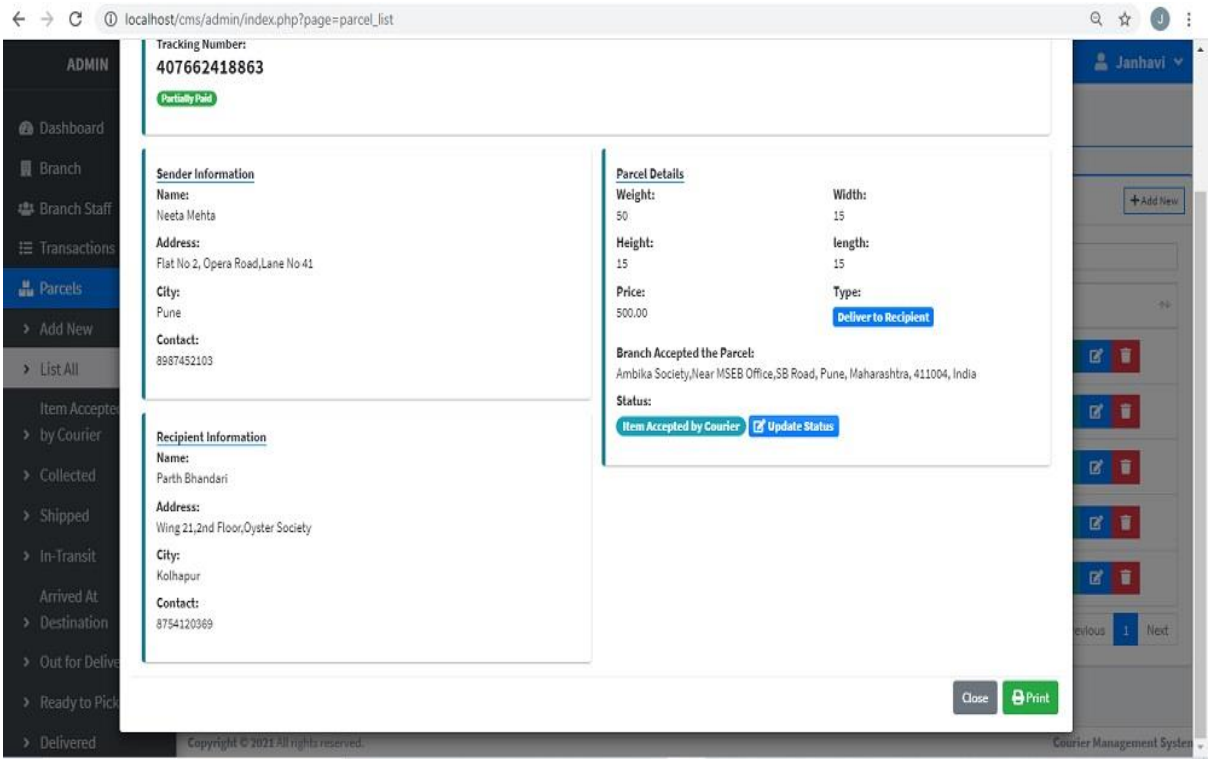
Parcel List


localhost/cms/admin/index.php?page=parcel_list
ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Transactions
Parcels
Add New
List All
Item Accepted by Courier
Collected
Shipped
In-Transit
Arrived At Destination
Out for Delivery
Ready to Pickup
Delivered
Parcel List
+Add New
Show 10 entries
Search:
Reference Number Sender Name Recipient Name Status Action
1 407662418863 Neeta Mehta Parth Bhandari Item Accepted by Courier
2 088153818493 Neeta Mehta Parth Bhandari Item Accepted by Courier
3 658364359519 Karan Patel Karishma Limaye Collected
4 923924427303 Mayuri Bhosle Akansha Sharma Out for Delivery
5 512645106699 Unnati Waingankar Dipali Kadam Collected
Showing 1 to 5 of 5 entries
Previous 1 Next
Copyright © 2021 All rights reserved.
Courier Management System

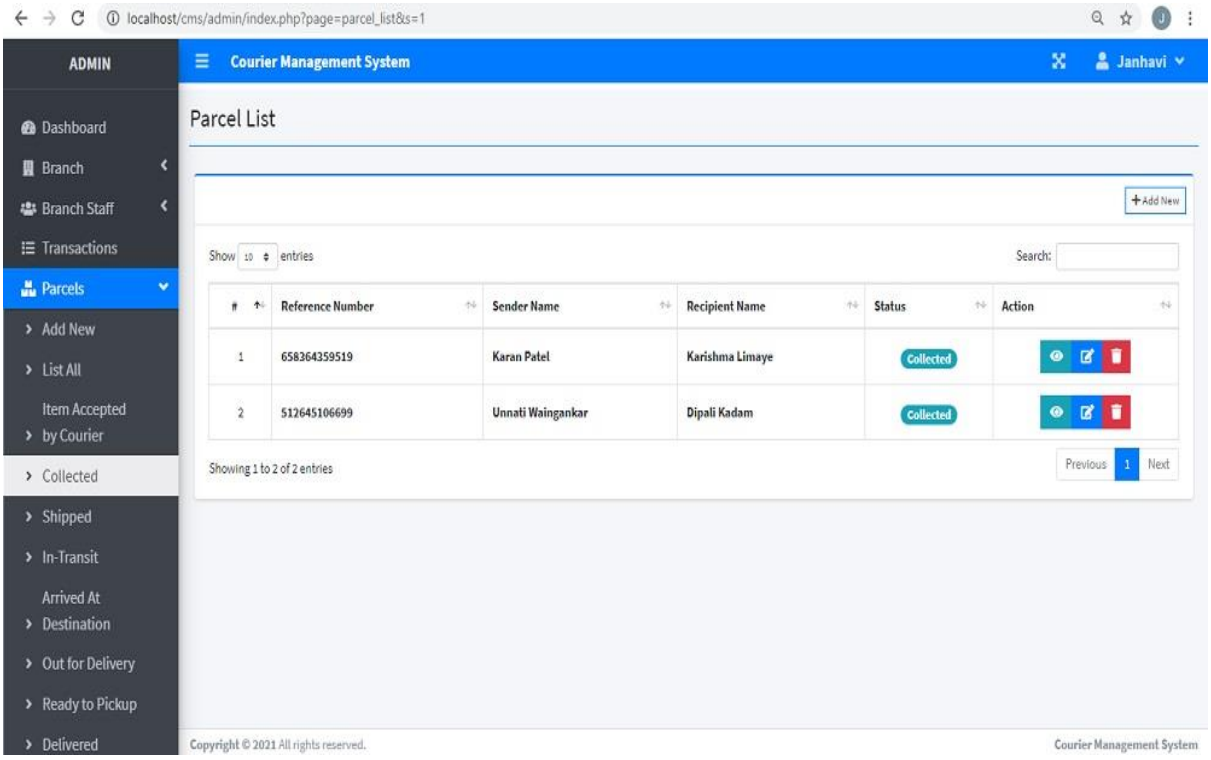
View Parcel Details



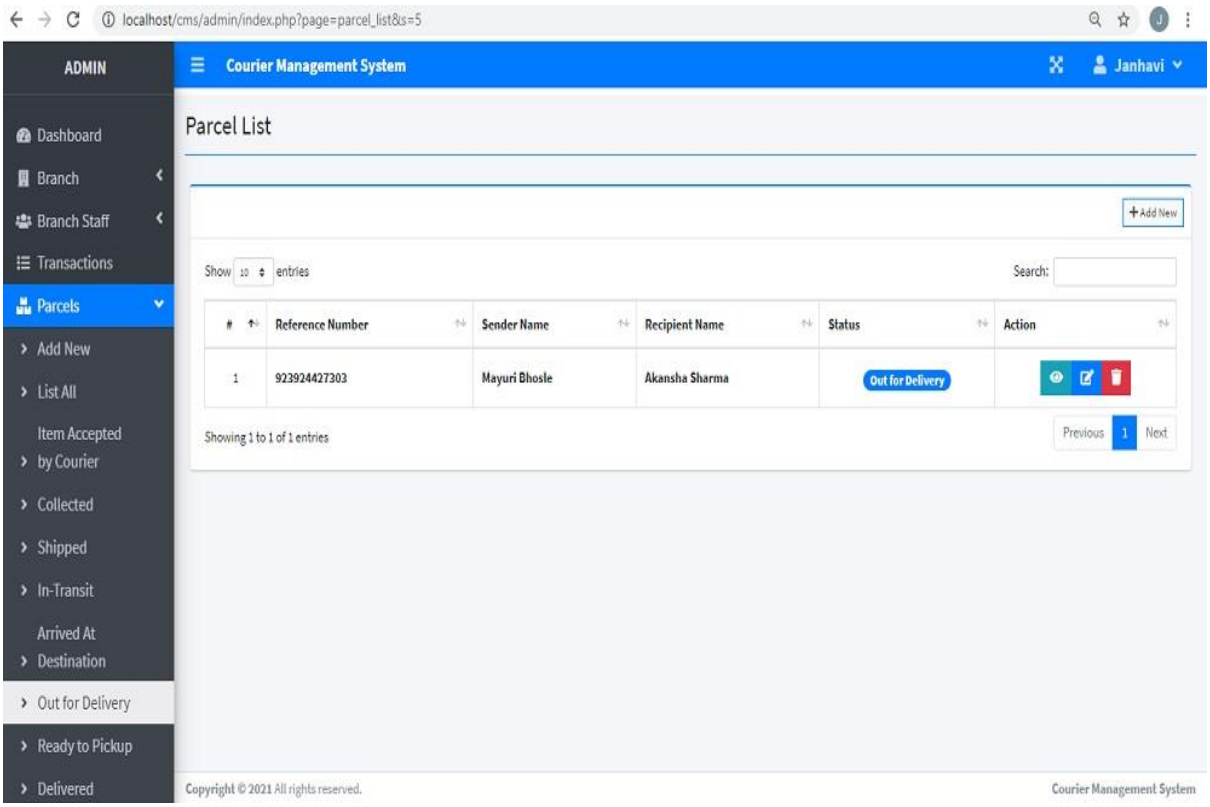
View Parcel Details



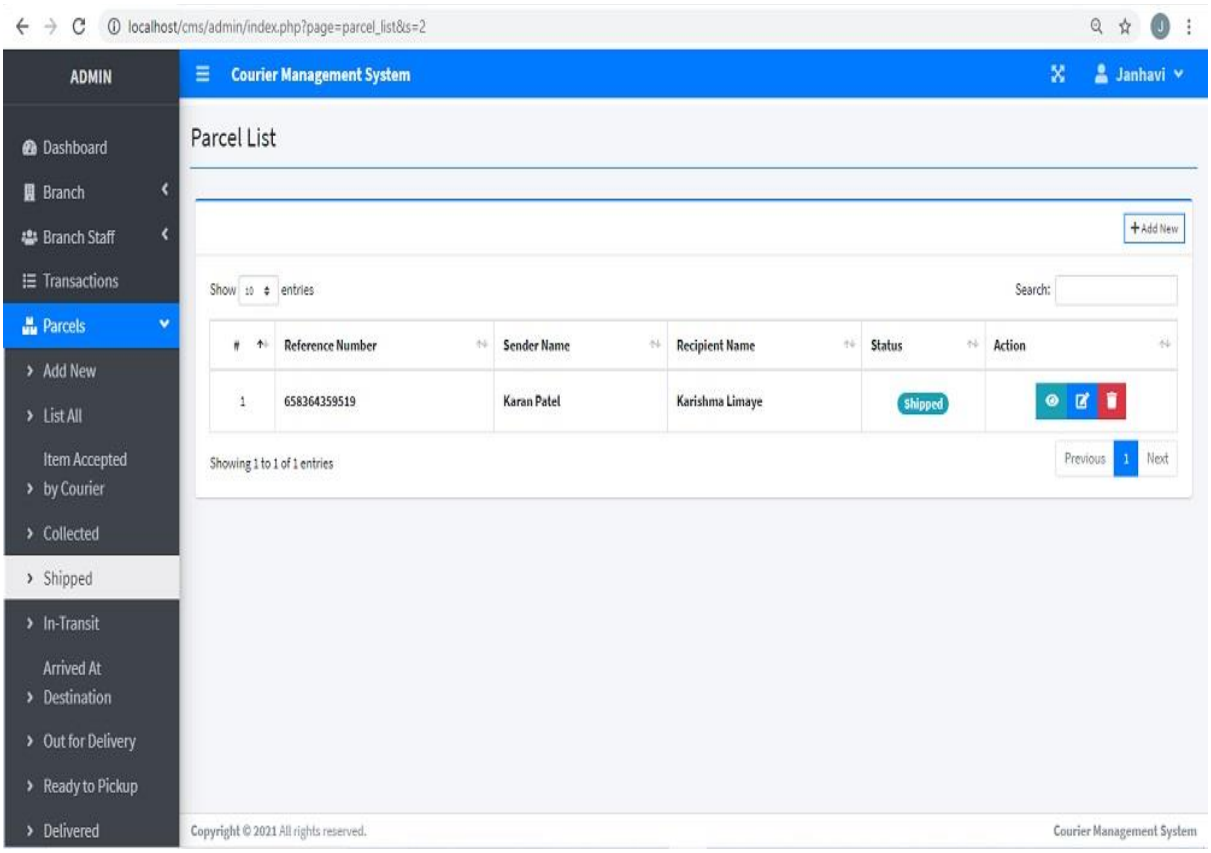
Courier Status



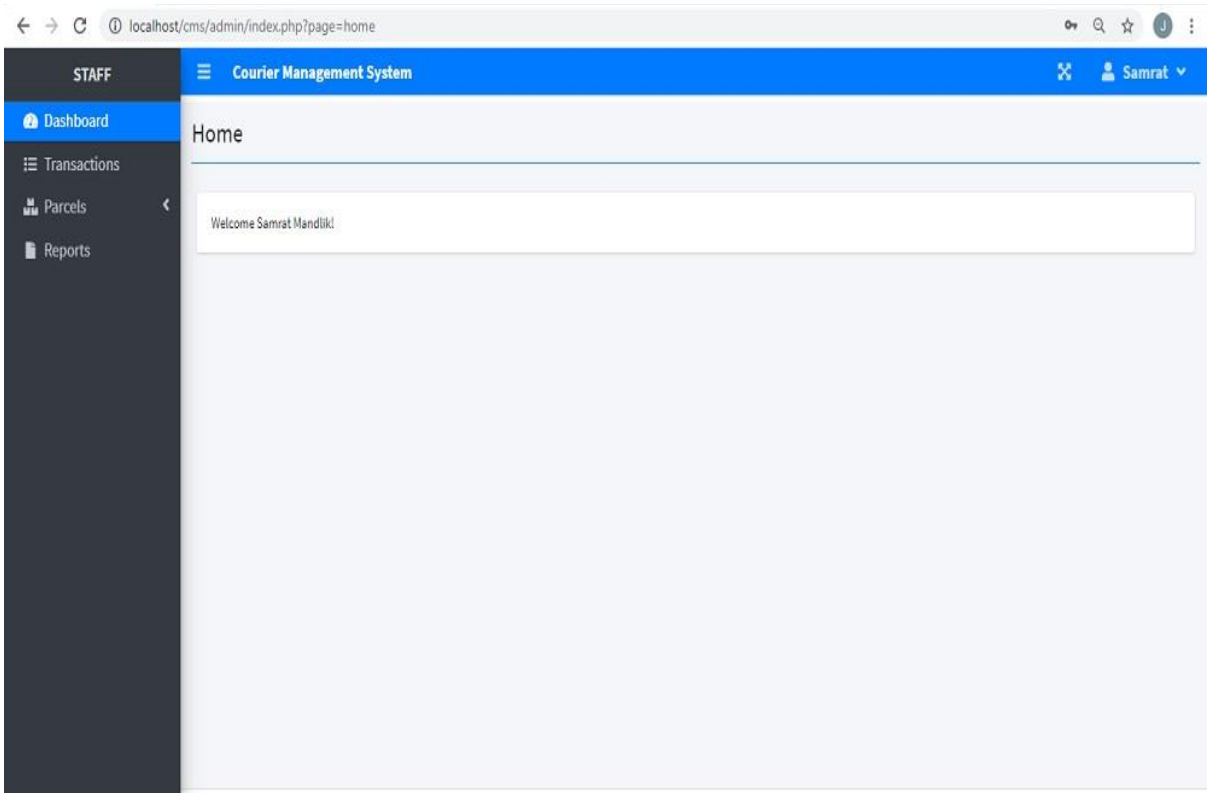
Courier Status



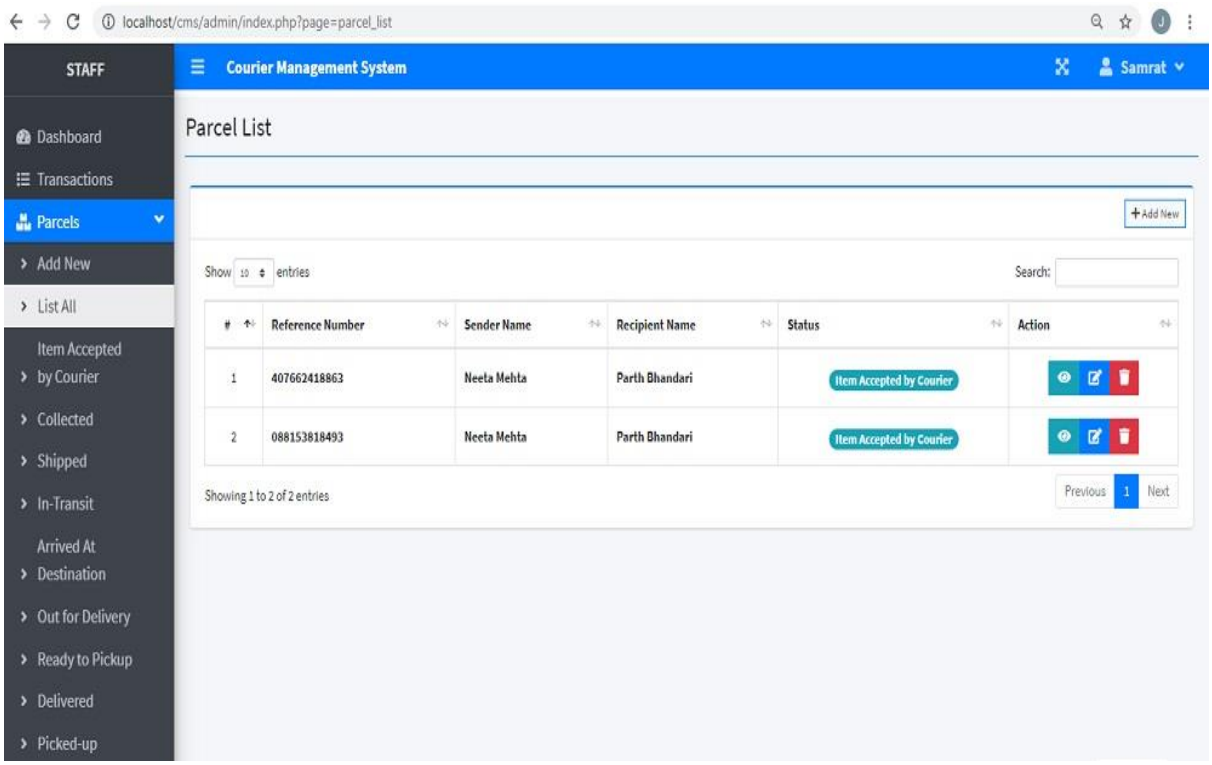
Courier Status



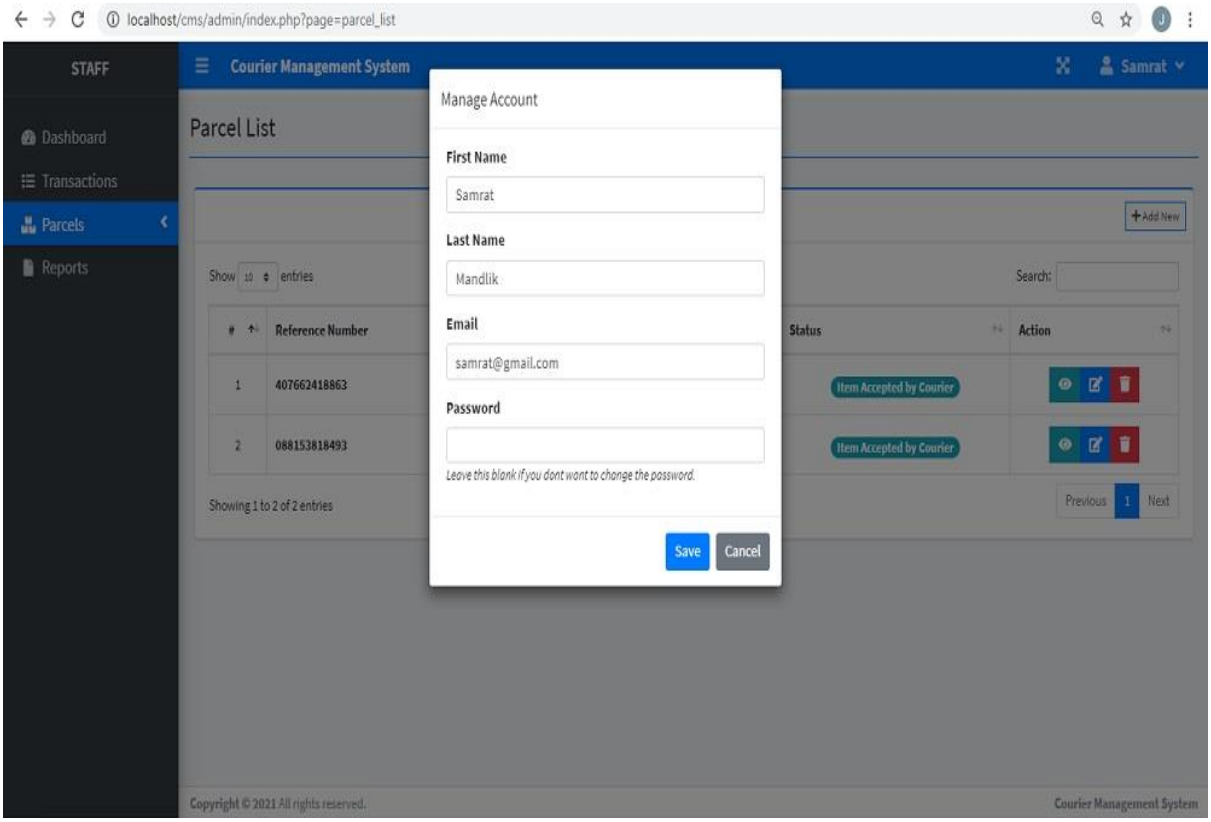
Staff Dashboard



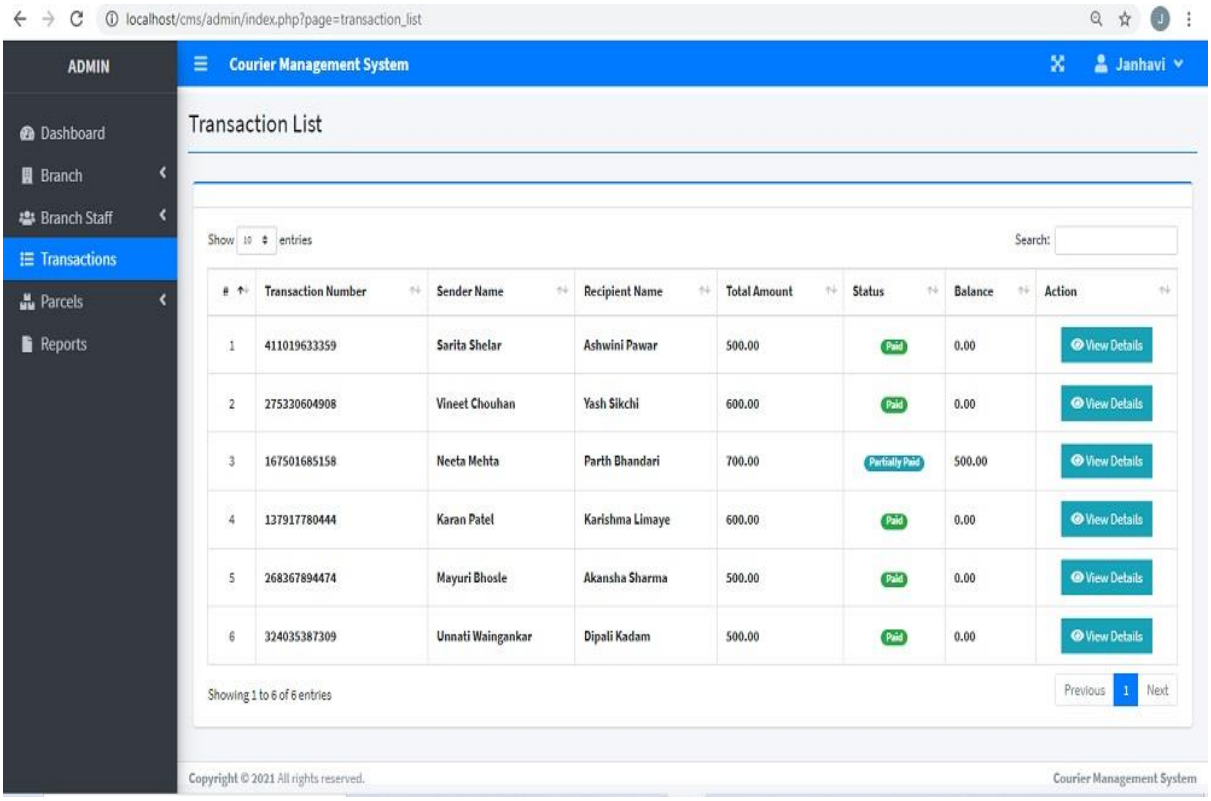
Staff Parcel List



Update Staff



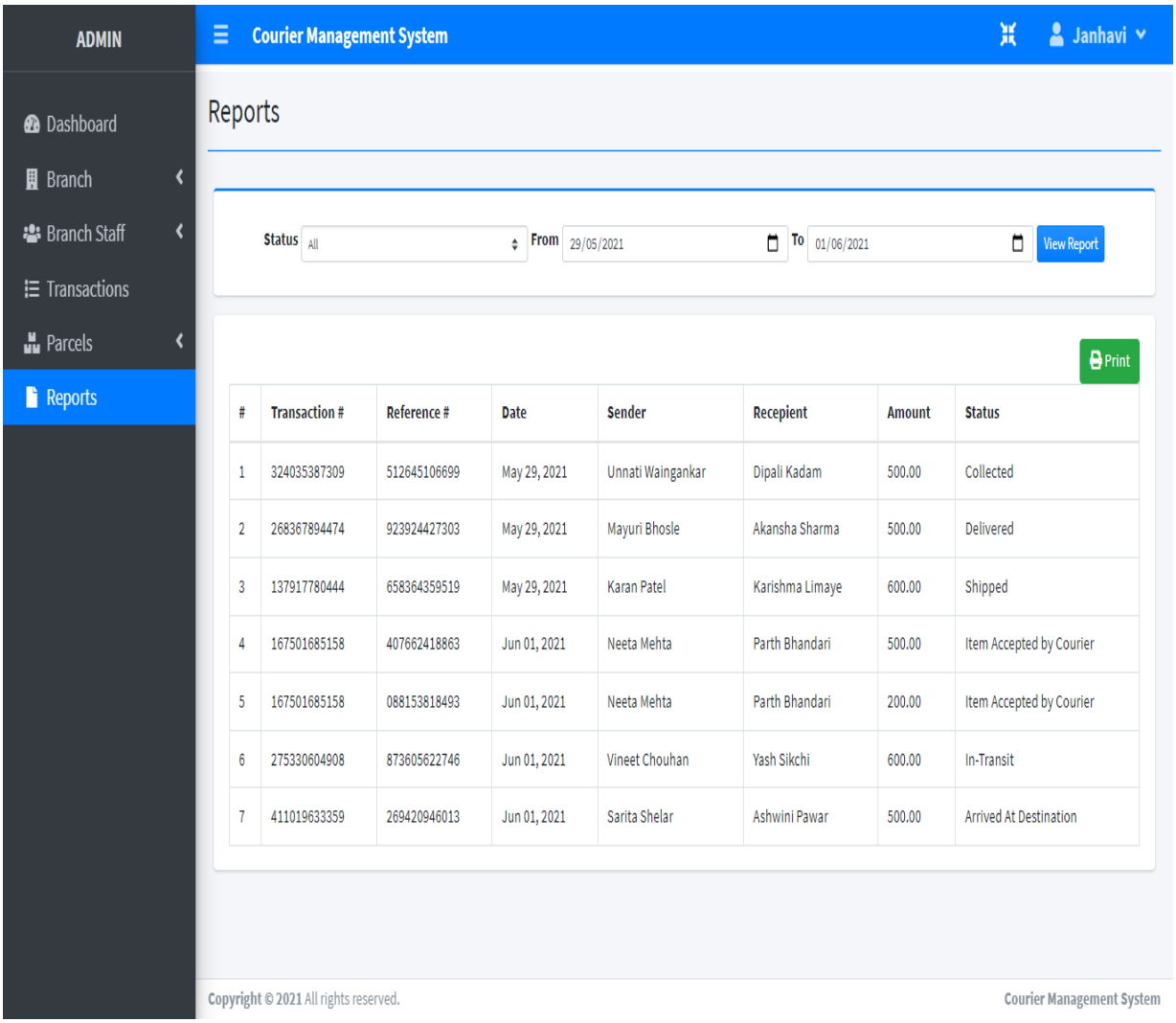
Transactions

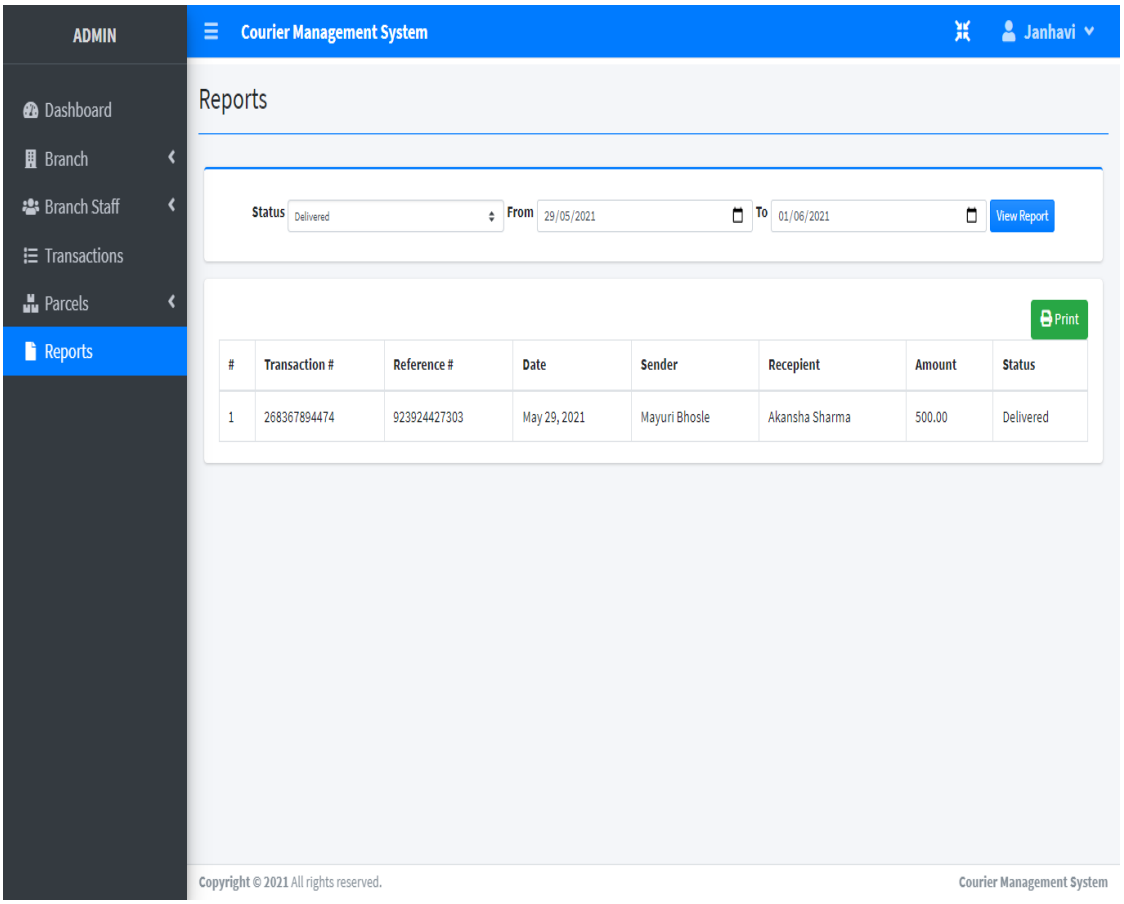


ANNEXURE 2

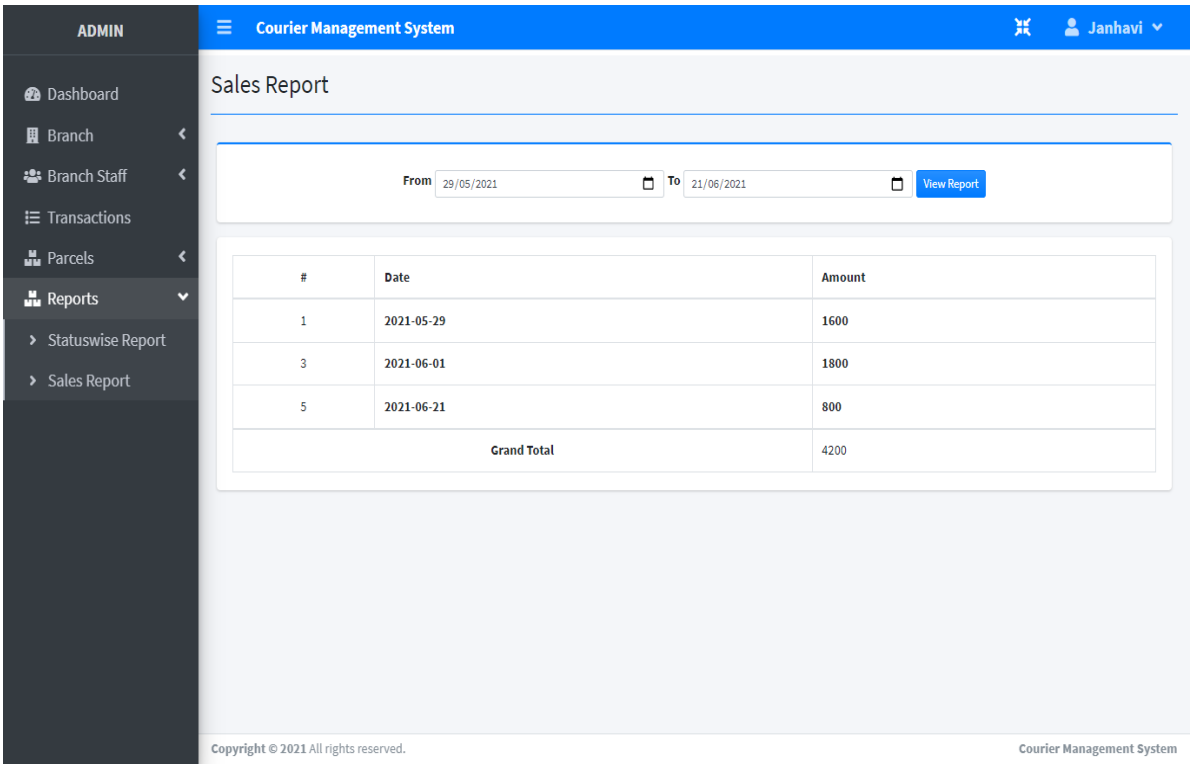
OUTPUT REPORTS

Datewise and statuswise Courier Reports




ADMIN
Courier Management System
Janhavi
Dashboard
Branch
Branch Staff
Transactions
Parcels
Reports
Reports
Status
Delivered
From
29/05/2021
To
01/06/2021
View Report
Print
Transaction # Reference # Date Sender Recepient Amount Status
1 268367894474 923924427303 May 29, 2021 Mayuri Bhosle Akansha Sharma 500.00 Delivered
Copyright © 2021 All rights reserved.
Courier Management System

Courier Sales Report



ANNEXURE 3

SAMPLE PROGRAM CODE

Code for adding Courier

```
<?php if(!isset($conn)){ include 'db_connect.php'; } ?>
<style>
  textarea{
    resize: none;
  }
</style>
<div class="col-lg-12">
          <div class="card card-outline card-primary">
                  <div class="card-body">
                          <form action="" id="manage-parcel" name="add_parcel">
      <input type="hidden" name="id" value="<?php echo isset($id) ? $id : '' ?>">
      <input type="hidden" name="pd_id" value="<?php echo isset($pd_id) ? $pd_id : '' ?>">
      <div id="msg" class=""></div>
      <div class="row">
        <div class="col-md-6">
            <b style="font-size: 18px;">Sender Information</b>
```

```
<div class="form-group">
 <label style="font-size: 18px;" for="" class="control-label">Full Name</label>
 <input style="font-size: 18px;"type="text" name="sender_name" id="sender_name" class="form-control form-control-sm" oninput="senderName_Validate()" value="<?php echo isset($sender_name) ? $sender_name : " ?>" required>
 <p id="senderName" style="color: red"></p>
</div>
<div class="form-group">
 <label style="font-size: 18px;" for="" class="control-label">Address</label>
 <input style="font-size: 18px;" type="text" name="sender_address" id="" class="form-control form-control-sm" value="<?php echo isset($sender_address) ? $sender_address : " ?>" required>
</div>
<div class="form-group">
 <label style="font-size: 18px;" for="" class="control-label">City</label>
```

```
            <input style="font-size: 18px;" type="text"
name="sender_city" id="sender_city" class="form-control form-control-
sm" oninput="senderCity_Validate()" value="<?php echo
isset($sender_city) ? $sender_city : '' ?>" required>
             <p id="senderCity" style="color: red"></p>
            </div>
            <div class="form-group">
             <label style="font-size: 18px;" for="" class="control-
label">Contact </label>
             <input style="font-size: 18px;" type="text"
name="sender_contact" id="sender_contact" class="form-control form-
control-sm" oninput="senderContact_Validate()" value="<?php echo
isset($sender_contact) ? $sender_contact : '' ?>" required>
             <p id="senderContact" style="color: red"></p>
            </div>
        </div>
        <div class="col-md-6">
            <b style="font-size: 18px;">Recipient Information</b>
            <div class="form-group">
```

```
<label style="font-size: 18px;" for="" class="control-label">Full Name</label>
<input style="font-size: 18px;" type="text" name="recipient_name" id="recipient_name" class="form-control form-control-sm" oninput="recipientName_Validate() " value="<?php echo isset($recipient_name) ? $recipient_name : " ?>" required>
<p id="recipientName" style="color: red"></p>
</div>
<div class="form-group">
<label style="font-size: 18px;" for="" class="control-label">Address</label>
<input style="font-size: 18px;" type="text" name="recipient_address" id="" class="form-control form-control-sm" value="<?php echo isset($recipient_address) ? $recipient_address : " ?>" required>
</div>
<div class="form-group">
<label style="font-size: 18px;" for="" class="control-label">City</label>
```

```
            <input style="font-size: 18px;" type="text" name="recipient_city" id="recipient_city" class="form-control form-control-sm" oninput = "recipientCity_Validate()"value="<?php echo isset($recipient_city) ? $recipient_city : " ?>" required>
             <p id="recipientCity" style="color: red"></p>
            </div>
            <div class="form-group">
             <label style="font-size: 18px;" for="" class="control-label">Contact </label>
             <input style="font-size: 18px;" type="text" name="recipient_contact" id="recipient_contact" class="form-control form-control-sm" oninput="recipientContact_Validate()"value="<?php echo isset($recipient_contact) ? $recipient_contact : " ?>" required>
             <p id="recipientContact" style="color: red"></p>
            </div>
          </div>
        </div>
        <hr>
        <div class="row">
          <div class="col-md-6">
```

```
<div class="form-group">
  <label style="font-size: 18px;" for="dtype">Type</label>
  <input style="font-size: 18px;" type="checkbox" name="type" id="dtype" <?php echo isset($type) && $type == 1 ? 'checked' : '' ?> data-bootstrap-switch data-toggle="toggle" data-on="Deliver" data-off="Pickup" class="switch-toggle status_chk" data-size="xs" data-offstyle="info" data-width="5rem" value="1">
  <small style="font-size: 15px;">Deliver = Deliver to Recipient Address</small>
  <small style="font-size: 15px;">, Pickup = Pickup to nearest Branch</small>
 </div>
</div>
<div class="col-md-6" id=""  <?php echo isset($type) && $type == 1 ? 'style="display: none"' : '' ?>>
  <?php if($_SESSION['login_branch_id'] <= 0): ?>
  <div class="form-group" id="fbi-field">
   <label style="font-size: 18px;" for="" class="control-label">Branch Processed</label>
```

```
            <select name="from_branch_id" id="from_branch_id" class="form-control select2" required="">
              <option style="font-size: 18px;" value=""></option>
              <?php
               $branches = $conn->query("SELECT *,concat(street,', ',city,', ',state,', ',pincode,', ',country) as address FROM branches ");
                 while($row = $branches->fetch_assoc()):
              ?>
               <option value="<?php echo $row['id'] ?>" <?php echo isset($from_branch_id) && $from_branch_id == $row['id'] ? "selected":" ?>><?php echo $row['branch_code']. ' | '.(ucwords($row['address'])) ?></option>
              <?php endwhile; ?>
            </select>
          </div>
          <?php else: ?>
           <input type="hidden" name="from_branch_id" value="<?php echo $_SESSION['login_branch_id'] ?>">
          <?php endif; ?>
          <div class="form-group" id="tbi-field">
```

```
                <option style="font-size:18px;" value=""></option>
              <label style="font-size: 18px;" for="" class="control-label">Pickup Branch</label>
              <select name="to_branch_id" id="to_branch_id" class="form-control select2">
                <option value=""></option>
                <?php
                  $branches = $conn->query("SELECT *,concat(street,', ',city,', ',state,', ',pincode,', ',country) as address FROM branches");
                    while($row = $branches->fetch_assoc()):
                ?>
                  <option value="<?php echo $row['id'] ?>" <?php echo isset($to_branch_id) && $to_branch_id == $row['id'] ? "selected":"" ?>><?php echo $row['branch_code']. ' | '.(ucwords($row['address'])) ?></option>
                <?php endwhile; ?>
              </select>
            </div>
          </div>
        </div>
```

```
<hr>
<b style="font-size:18px;">Parcel Information</b>
<table class="table table-bordered" id="parcel-items">
  <thead>
    <tr style="font-size:18px;">
      <th>Weight(in kg)</th>
      <th>Height(in cms)</th>
      <th>Length(in cms)</th>
      <th>Width(in cms)</th>
      <th>Price</th>
      <?php if(!isset($id)): ?>
      <th></th>
    <?php endif; ?>
    </tr>
  </thead>
  <tbody>
    <tr style="font-size:18px;">
      <td><input type="text" name='weight[]' value="<?php echo isset($weight) ? $weight :" ?>" required placeholder="in kgs"></td>
```

```
            <td><input type="text" name='height[]' value="<?php echo isset($height) ? $height :" ?>" required placeholder="in cms"></td>
            <td><input type="text" name='length[]' value="<?php echo isset($length) ? $length :" ?>" required placeholder="in cms"></td>
            <td><input type="text" name='width[]' value="<?php echo isset($width) ? $width :" ?>" required placeholder="in cms"></td>
            <td><input type="text" name='price[]' value="<?php echo isset($price) ? $price :" ?>" required ></td>
            <?php if(!isset($id)): ?>
            <td><button class="btn btn-sm btn-danger" type="button" onclick="$(this).closest('tr').remove() && calc()"><i class="fa fa-times"></i></button></td>
            <?php endif; ?>
          </tr>
        </tbody>
            <?php if(!isset($id)): ?>
        <tfoot style="font-size:18px;">
          <th colspan="4" class="text-right">Total</th>
          <th class="text-right" id="tAmount">0.00</th>
          <th></th>
```

```
  </tfoot>
  <input type="hidden" id="amount" name="amount">
     <?php endif; ?>
 </table>
     <?php if(!isset($id)): ?>
 <div class="row">
  <div class="col-md-6">
   <div class="form-group">
    <label style="font-size:18px;" for="">Amount Paid</label>
    <input style="font-size:18px;" type="text" name="payment" id="" class="form-control form-control-sm" value="<?php echo isset($payment) ? $payment : '' ?>" required value="0">
   </div>
  </div>
  <div class="col-md-12 d-flex justify-content-end">
   <button style="font-size:18px;"  class="btn btn-sm btn-primary bg-gradient-primary" type="button" id="new_parcel"><i class="fa fa-item"></i> Add Item</button>
  </div>
 </div>
```

```
            <?php endif; ?>
        </form>
            </div>
            <div class="card-footer border-top border-info">
                    <div class="d-flex w-100 justify-content-center align-items-center">
                            <button style="font-size:18px;" class="btn btn-flat bg-gradient-primary mx-2" form="manage-parcel">Save</button>
                            <a style="font-size:18px;" class="btn btn-flat bg-gradient-secondary mx-2" href="./index.php?page=parcel_list">Cancel</a>
                    </div>
            </div>
            </div>
</div>
<div id="ptr_clone" class="d-none">
  <table>
    <tr style="font-size:18px;">
       <td><input type="text" name='weight[]' required></td>
       <td><input type="text" name='height[]' required></td>
```

```
        <td><input type="text" name='length[]' required></td>
        <td><input type="text" name='width[]' required></td>
        <td><input type="text" name='price[]' required ></td>
        <td><button class="btn btn-sm btn-danger" type="button" onclick="$(this).closest('tr').remove() && calc()"><i class="fa fa-times"></i></button></td>
      </tr>
  </table>
</div>
<script>
  function senderName_Validate() {
    var x = document.getElementById("sender_name").value;
    if(!/^[a-zA-Z-,]+(\s{0,1}[a-zA-Z-,
])*$/g.test(document.add_parcel.sender_name.value)){
      document.getElementById("senderName").innerHTML = "Please enter alphabets or characters only";
      document.add_parcel.sender_name.focus();
    }
    else{
      document.getElementById("senderName").innerHTML = " ";
```

```
    }
  }
  function recipientName_Validate() {
    var x = document.getElementById("recipient_name").value;
    if(!/^[a-zA-Z-,]+(\s{0,1}[a-zA-Z-,
])*$/g.test(document.add_parcel.recipient_name.value)){
      document.getElementById("recipientName").innerHTML  =  "Please
enter alphabets or characters only";
      document.add_parcel.recipient_name.focus();
    }
    else{
      document.getElementById("recipientName").innerHTML = " ";
    }
  }
  function senderCity_Validate() {
    var x = document.getElementById("sender_city").value;
    if(!/^[a-zA-Z-,]+(\s{0,1}[a-zA-Z-,
])*$/g.test(document.add_parcel.sender_city.value)){
      document.getElementById("senderCity").innerHTML     =     "Please
enter alphabets or characters only";
```

```
      document.add_parcel.sender_city.focus();
    }
    else{
      document.getElementById("senderCity").innerHTML = " ";
    }
  }
  function recipientCity_Validate() {
    var x = document.getElementById("recipient_city").value;
    if(!/^[a-zA-Z-,]+(\s{0,1}[a-zA-Z-,
])*$/g.test(document.add_parcel.recipient_city.value)){
      document.getElementById("recipientCity").innerHTML   =   "Please
enter alphabets or characters only";
      document.add_parcel.recipient_city.focus();
    }
    else{
      document.getElementById("recipientCity").innerHTML = " ";
    }
  }
  function senderContact_Validate() {
    var y = document.getElementById('sender_contact').value;
```

```
    if (/^\d{11}$/.test(y)) {
      document.getElementById("senderContact").innerHTML = "Enter
upto 10 numbers only";
    }
    else {
      document.getElementById("senderContact").innerHTML = " ";
    }
  }
  function recipientContact_Validate() {
    var y = document.getElementById('recipient_contact').value;
    if (/^\d{11}$/.test(y)) {
      document.getElementById("recipientContact").innerHTML = "Enter
upto 10 numbers only";
    }
    else {
      document.getElementById("recipientContact").innerHTML = " ";
    }
  }
  /*function calcTotal() {
    var msg;
```

```
  var weight = parseInt($('[name="weight[]"]').value; );
  var discount;
  var total = $('#parcel-items [name="price[]"]').text(parseFloat(total))
  if( weight >= 0 && weight <= 150 ) {
    total = weight * 20
  }
  else if( weight >150 && weight <= 300 ) {
    total = weight * 15
  }
  else if( weight >300 && weight <= 400 ) {
    total = weight * 10

  }
}*/

$('#dtype').change(function(){
    if($(this).prop('checked') == true){
      $('#tbi-field').hide()
    }else{
      $('#tbi-field').show()
```

```
    }
})
  $('[name="price[]"]').keyup(function(){
    calc()
  })
$('#new_parcel').click(function(){
  var tr = $('#ptr_clone tr').clone()
  $('#parcel-items tbody').append(tr)
  $('[name="price[]"]').keyup(function(){
    calc()
  })
  $('.number').on('input keyup keypress',function(){
      var val = $(this).val()
      val = val.replace(/[^0-9]/, '');
      val = val.replace(/,/g, '');
      val = val > 0 ? parseFloat(val).toLocaleString("en-US") : 0;
      $(this).val(val)
  })

})
```

```
	$('#manage-parcel').submit(function(e){
		e.preventDefault()
		start_load()
if($('#parcel-items tbody tr').length <= 0){
  alert_toast("Please add atleast 1 parcel information.","error")
  end_load()
  return false;
}
		$.ajax({
			url:'ajax.php?action=save_parcel',
			data: new FormData($(this)[0]),
		    cache: false,
		    contentType: false,
		    processData: false,
		    method: 'POST',
		    type: 'POST',
			success:function(resp){
			// if(resp){
  //       resp = JSON.parse(resp)
  //       if(resp.status == 1){
```

```
//          alert_toast('Data successfully saved',"success");
//          end_load()
//                                                    var        nw        =
window.open('print_pdets.php?ids='+resp.ids,"_blank","height=700,wid
th=900")
//       }
                              // }
       if(resp == 1){
          alert_toast('Data successfully saved',"success");
          setTimeout(function(){
            location.href = 'index.php?page=parcel_list';
          },2000)

       }else{
          alert_toast('An error occured',"error");
          end_load()
       }
                              }
                    })
          })
```

```
function displayImgCover(input,_this) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
          $('#cover').attr('src', e.target.result);
        }

        reader.readAsDataURL(input.files[0]);
    }
}
function calc(){

      var total = 0 ;
       $('#parcel-items [name="price[]"]').each(function(){
        var p = $(this).val();
            p =  p.replace(/,/g,'')
            p = p > 0 ? p : 0;
          total = parseFloat(p) + parseFloat(total)
       })
       if($('#tAmount').length > 0)
```

```
        $('#tAmount').text(parseFloat(total).toLocaleString('en-
US',{style:'decimal',maximumFractionDigits:2,minimumFractionDigits:
2}))
        $('#amount').val(parseFloat(total))
  }
</script>
```