VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A DBMS Mini-Project Report
On

"Courier Database Management System"

Submitted in partial fulfillment of the requirements for the 5th semester of
Bachelor of Engineering in Computer Science and Engineering
of Visvesvaraya Technological University, Belagavi

Submitted by:

KESHAV DAGA	**1RN19CS064**
ROUNAK AGARWAL	**1RN19CS116**

Under the Guidance of:

Mrs.Soumya N G	**Mr.Sanjay P K**
Assistant Professor	**Assistant Professor**
Dept. of CSE	**Dept. of CSE**

Department of Computer Science and Engineering
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2021-2022

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road,
Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the DBMS mini-project work entitled **"Courier Database Management System"** has been successfully carried out by **Keshav Daga** bearing USN **1RN19CS064** and **Rounak Agarwal** bearing USN **1RN19CS116**, bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **5th semester Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the mini-project requirements of the DBMS lab of 5th semester BE in CSE.

Mrs.Soumya N G	**Mr.Sanjay P K**	**Dr. Kiran P**
Assistant Professor	**Assistant Professor**	**Prof. and Head**
Dept. of CSE	**Dept. of CSE**	**Dept of CSE**

External Viva:

Name of the Examiners	**Signature with Date**
1.	
2.	

ACKNOWLEDGMENTS

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to **Dr. M K Venkatesha,** Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. Kiran P** Prof. & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our **Mrs. Soumya N G,** Assistant Professor, **Mr. Sanjay P K**, Assistant Professor, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date	:	Keshav Daga	1RN19CS064
Place	: Bengaluru	Rounak Agarwal	1RN19CS116

ABSTRACT

The COURIER DATABASE MANAGEMENT SYSTEM is a web based system that' designed primarily for the use in the couriers logistics industry this system will allow courier and logistical services company to increase scope of the business by reducing the paper work cost and accountability of goods involved this system also allows quick and easy management of transporting parcels from one point to another as they can be easily tracked compared to the use of manual systems of recording information as it includes message sent to the receiver and the sender to track the parcel .courier services employees use the system through an easy to navigate graphical interface for efficient processing. This project is developed using HTML, CSS, PHP and MYSQL. This system is an online application which can be hosted online and therefore the user needs an internet connection or the company's local area network.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 DATABASE TECHNOLOGIES

The essential feature of database technology is that it provides an internal representation (model) of the external world of interest. Examples are the representation of a particular date/time/flight/aircraft in airline reservation or of item code/item description/quantity on hand/reorder level/reorder quantity in a stock control system.
The technology involved is concerned primarily with maintaining the internal representation consistent with external reality; this involves the results of extensive R&D over the past 30 years in areas such as user requirements analysis, data modelling, process modelling, data integrity, concurrency, transactions, file organization, indexing, rollback and recovery, persistent programming, object-orientation, logic programming, deductive database systems, active database systems... and in all these (and other) areas there remains much to be done.
The essential point is that database technology is a CORE TECHNOLOGY with links to:

- Information management / processing
- Data analysis / statistics
- Data visualization / presentation
- Multimedia and hypermedia
- Office and document systems
- Business processes, workflow, CSCW (computer-supported cooperative work)

Relational DBMS is the modern base technology for many business applications. It offers flexibility and easy-to-use tools at the expense of ultimate performance. More recently relational systems have started to extend their facilities in the directions of information retrieval, object-orientation and deductive/active systems leading to the so-called 'Extended Relational Systems'.

Information Retrieval Systems began with handling library catalogues and then extended to full free-text utilizing inverted index technology with a lexicon or thesaurus. Modern systems utilize some KBS (knowledge-based systems) techniques to improve retrieval.

Object-Oriented DBMS started for engineering applications where objects are complex, have versions and need to be treated as a complete entity. OODBMSs share many of the OOPL features such as identity, inheritance, late binding, overloading and overriding.

OODBMSs have found favours in engineering and office systems but have not yet been successful in traditional application areas.

Deductive / Active DBMS have emerged over the last 20 years and combine logic programming technology with database technology. This allows the database itself to react to external events and to maintain dynamically its integrity with respect to the real world.

1.2 CHARACTERISTICS OF DATABASE APPROACH

Traditionally, data was organized in file formats. DBMS was a new concept then, and research was done to make it overcome the deficiencies of traditional style of data management. A modern DBMS has the following characteristics −

- Real-world entity − A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.

- Relation-based tables − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.

- Isolation of data and application − A database system is entirely different from its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about the data, to ease its own process.

- Less redundancy − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.

- Consistency − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency when compared to earlier forms of data storing applications like file-processing systems.

- Query Language − DBMS is equipped with query language, which makes it very efficient in retrieving and manipulating data. A user can apply as many and as different filtering options as required to retrieve a set of data. This was not possible when file-processing system was used.

- ACID Properties − DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.

- Multiuser and Concurrent Access − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when user attempts to handle the same data item, the users are always unaware of them.

- Multiple views − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of the database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.

- Security − Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS provides methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

1.3 APPLICATIONS OF DBMS

Applications of Database Management Systems are:

- **Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain the huge amount of data that keeps getting updated every millisecond.
- **Industry**: Whether it is a manufacturing unit, or a warehouse or distribution center, each one needs a database to keep the records of ins and outs. For example, distribution center should keep a track of the product units that supplied into the

centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.

- **Banking System**: For storing customer info, tracking his/her day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
- **Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is lots of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online shopping**: You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

1.4 PROBLEM DESCRIPTION/STATEMENT

In case of manual system, they need a lot of time, manpower etc. Here almost every aspect of work is computerized. So that accuracy is maintained. Courier Management System mainly deals with Customer Profile, Orders, Contact and Tracking information which can be managed by the Admin.

Computerized Courier DBMS is developed to facilitate the general administration to manage and swift, efficient and delivery of parcels without delay. So that, on the company side profit can be increased and facilitates better control. And for customer fast, accurate service can be provided.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 HARDWARE REQUIREMENTS

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor	:	Pentium4 processor
Processor Speed	:	2.4 GHz
RAM	:	1 GB
Storage Space	:	40 GB
Monitor Resolution	:	1024*768 or 1336*768 or 1280*1024

2.2 SOFTWARE REQUIREMENTS

- Operating System used: Windows10
- Visual Studio Code Editor: HTML, CSS, JavaScript, PHP
- Apache XAMPP Server: MySQL, PhpMyAdmin
- IDE used: Visual Studio Code
- Browser that supports HTML

2.3 FUNCTIONAL REQUIREMENTS

2.3.1 Major Entities

Admin: The admin has these functions

- View all records
- Facilitate timely and fast delivery
- Responding to grievance

Customer: Customer is the entity who is going to place a shipment. Few attributes are id, name, phone, email etc.

2.3.2 End User Requirements

The technical requirements for the project are mentioned below.

2.3.3 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from a local storage and render them to multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects like interactive forms can be embedded into the rendered page. It provides a way to create structured documents by denoting structural semantics for the text like headings, paragraphs, lists, links, quotes and other items. HTML elements are delimited by tags that are written within angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Other tags such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can also embed programs written in a scripting language such as JavaScript which affect the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content.

2.3.4 CSS

Cascading Style Sheets (CSS) is a style sheet language which is used for describing the presentation of a document written in a markup language. Although most often its 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 plain XML, SVG and XUL, and is also 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, colours, 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 the formatting by specifying the

relevant CSS in a separate css file, and reduce complexity and repetition in the structural content.

2.3.5 PHP

PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Pre-processor.

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

The standard PHP interpreter, powered by the Zend Engine, is a free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers, on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone into creating a formal PHP specification. HP development began in 1995 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used in order to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic web applications. To accelerate bug reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group on June 8, 1995 This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.

2.3.6 MySql

MySQL is a Relational Database Management System (RDBMS). MySQL server can manage many databases at the same time. In fact, many people might have different databases managed by a single MySQL server. Each database consists of a structure to hold the data. A database can exist without data as only a structure, be totally empty, twiddling its thumbs and waiting for data to be stored in it.

Data in a database is stored in one or more tables. You must create the database and the tables before you can add any data to the database. First create the empty database. Then you add empty tables to the database. Database tables are organized in rows and columns. Each row represents an entity in the database, such as a customer, a book, or a project. Each column contains an item of information about the entity, such as a customer name, a book name, or a project start date. The place where a particular row and column intersect i.e the individual cell of the table, is called a field. Tables in databases can be related. Often a row in one table is related to several rows in another table. For instance, you might have a database containing data about books you own. You would have a book table and an author table. One row in the author table might contain information about the author of several books in the book table. When tables are related, you include a column in one table to hold data that matches data in the column of another table.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

- MySQL is a database management system. A database is a structured collection of data. It can be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management system plays a central role in computing, as a standalone utilities, or as parts of other applications.
- MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query

Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and"SQL:2003" refers to the current version of the standard. We use the phrase "the SQLstandard" to refer to the current version of the SQL Standard at any time.

- MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. The MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- MySQL Server works on client/server or embedded systems. The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

2.3.7 XAMPP Server

Xampp server installs a complete, ready-to-use development environment. Xampp server allows you to fit your needs and allows you to setup a local server with the same characteristics as your production. While setting up the server and PHP on your own, you have two choices for the method of connecting PHP to the server. For many servers, PHP has a direct module interface (also called SAPI). These servers include Apache, Microsoft Internet Information Server, Netscape and iPlanet servers. Many other servers support ISAPI, the Microsoft module interface (OmniHTTPd for example). If PHP has no module support for your web server, you can always use it as a CGI or Fast CGI processor. This means you set up your server to use the CGI executable of PHP to process all PHP file requests on the server

CHAPTER 3

DATABASE DESIGN

3.1 Entities, Attributes and Relationships

Entities:

1.ADLOGIN

Email

Password

Aid

2.ADMIN

Aid

Phoneno

Email

Name

3.LOGIN

Email

Password

Uid

4.USERS

Uid

Name

Email

Phoneno

Aid

5.COURIER

cid

sname

rname

semail

remail

address

phoneno

billno

dates

sid

aid

6.CONTACTS

Email

Title

Comment

Date

uid

3.2 ER Schema

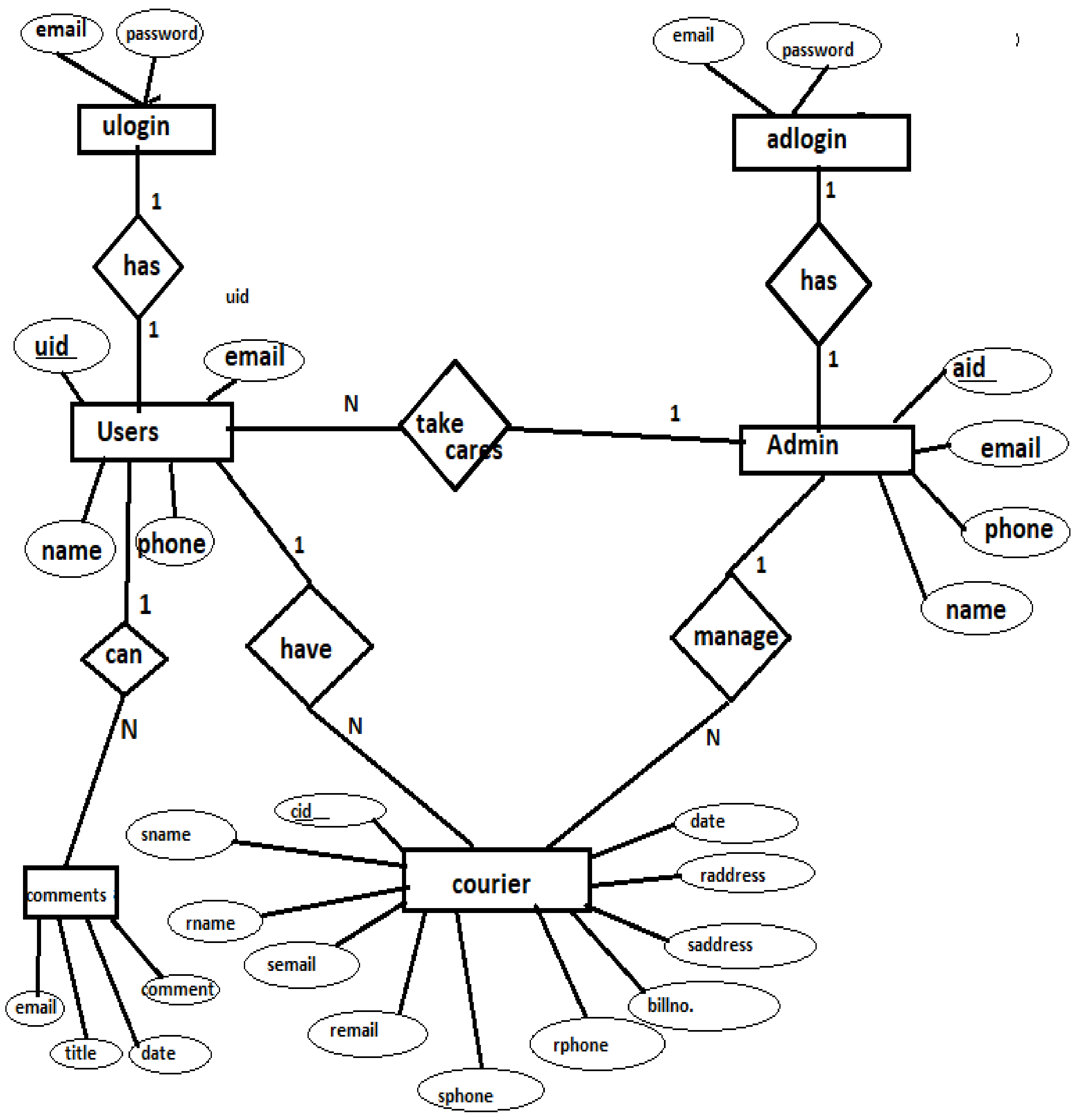


Fig 3.1 ER Diagram for Courier Database Management System

3.3 Relational Schema

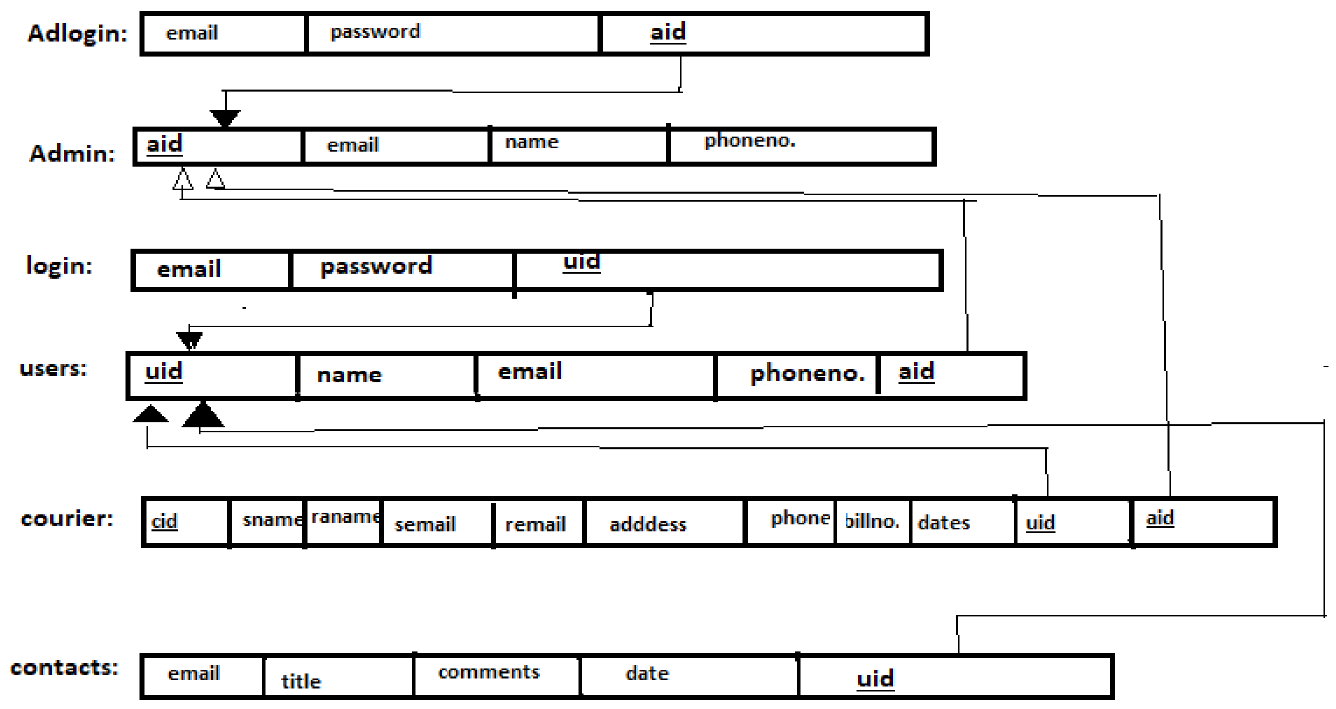


Fig 3.2 Relational Schema for Courier Database Management System

CHAPTER 4

IMPLEMENTATION

4.1 Creating DatabaseConnection

1. PHP provides built-in database connectivity for a wide range of databases – MySQL, PostgreSQL, Oracle, Berkeley DB, Informix, Lotus Notes, and more.

2.Useeithermysql_connectormysql_pconnecttocreatedatabaseconnection.

3.mysql_connect:connectionisclosedatendofscript(endofpage).

4. mysql_pconnect: creates persistent connection -connection remains even after endofthepage.

5.ConnecttotheMySQLserver

    5.1.$connection=mysql_connect("localhost",$username,$password);

6.Accessthedatabase

    6.1.mysql_select_db("databasename",$connection);

7.PerformSQLoperations

    Example:$result=mysql_query($query$connection)

8.Disconnectfromtheserver

    8.1mysql_close($connection);

4.2 ARCHITECTURE

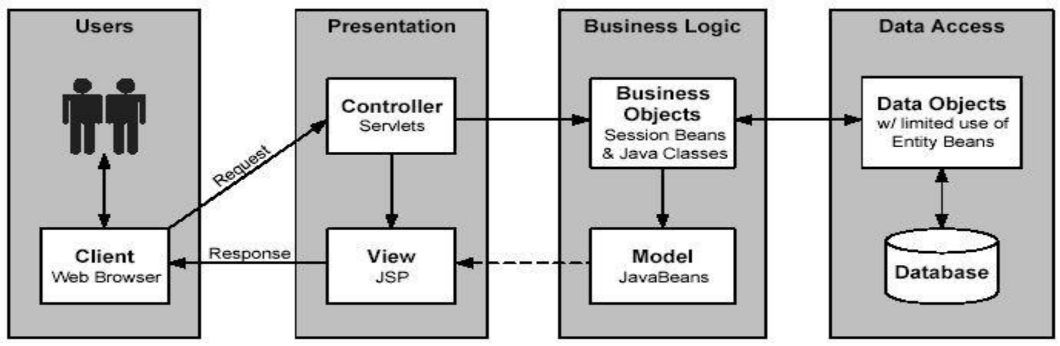


Fig.4.1. Architecture for Courier Database Management System

Four Tier architecture [Fig.4.1.] is a client–server architecture in which presentation, application processing, and data management functions are physically separated. Four-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application.

Presentation Layer

This is the topmost level of the application. The presentation tier displays information related to services such as browsing merchandise, purchasing and shopping cart contents. It also communicates with other tiers and puts out the results to the browser/client tier and to all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system's GUI)

Business Layer

Business layer or domain logic is the part of the program that encodes the real-world business rules which determine how data can be created, stored, and changed. It is contrasted with the remainder of the software that might be concerned with lower- level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program.

Data Access Layer

A Data Access Layer (DAL) in computer software, is a layer of computer program which provides simplified access to data stored in persistent storage. For example, the DAL might return a reference to an object (in terms of object-oriented programming) with its attributes instead of a row of fields from a database table. This allows the client (or user) modules to be created with a higher level of abstraction. This kind of model could be implemented by creating a class of data access methods that directly reference a corresponding set of database stored procedures. The DAL hides the complexity of the underlying data store from the external world.

Control Layer

The control layer is responsible for the communication between business and presentation layer. It connects logic and data with each other and provides a better connectivity and separation between layers.

4.3 Pseudo Code For Major Functionalities

Index

```
<?php
require_once "dbconnection.php";
require_once "session.php";
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {
  $email = $_POST['email'];
  $password = $_POST['password'];
  $qry = "SELECT * FROM `login` WHERE `email`='$email' AND
`password`='$password'";
  $run = mysqli_query($dbcon, $qry);
  $row = mysqli_num_rows($run);
  if ($row < 1) {
?>
    <script>
      alert("Opps! plz Enter Your Username and Pswd again..");
      window.open('index.php', '_self');
    </script> <?php
        } else {
          $data = mysqli_fetch_assoc($run);
          $id = $data['u_id'];    //fetch id value of user
          $email = $data['email'];
          $_SESSION['uid'] = $id;   //now we can use it until session destroy
          $_SESSION['emm'] = $email;
          ?>
    <script>
      alert("WELCOME ONBOARD");
      window.open('home/home.php', '_self');
      // changes made here
    </script> <?php
        }
      }
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <style>
        body {
            background-image: url('images/10.jpg');
            background-repeat: no-repeat;
            background-size: cover;
        }
    </style>
</head>
<body>
    <h1 align='center' style="margin: 15px; color:seagreen;font-weight: bold;font-
family:'Times New Roman', Times, serif">TYPHOON COURIER SERVICE</h1>
    <hr />
    <P align='center' style="font-weight: bold;color:orange;font-family:'Times New Roman',
Times, serif">The Fastest Courier Service Ever</P>
    <div>
        <h5><a href="admin/adminlogin.php" style="float: right; margin-right:40px; color:blue;
margin-top:0px">AdminLogin</a></h5>
    </div>
    <div class="container" style="margin-top: 60px; width:50%;">
        <div class="row">
            <div class="col-md-12">
                <h2 style="color: #273c75;">Login</h2>
                <p style="color:#e84118;">Please Fill Your ⤵⤵</p>
                <!-- <?php echo $error; ?> -->
            <form action="" method="post">
               <div class="form-group">
                  <label>Email Address</label>
                  <input type="email" name="email" class="form-control" placeholder="Enter
username/emailId" required />
               </div>
               <div class="form-group">
                  <label>Password</label>
                  <input type="password" name="password" class="form-control"
placeholder="Enter your password" required>
               </div>
               <div class="form-group">
                  <input type="submit" name="submit" class="btn btn-primary" value="SignIn"
/>
                  <button onclick="window.location.href='resetpswd.php'" class="btn btn-
danger" style="cursor:pointer">Reset Password</button>
               </div>
               <p style="color: #e84118;">Don't have an account?➤➤ <a
href="register.php">Register here</a>.</p>
            </form>
         </div>
      </div>
   </div>
</body>
</html>
```

Register

```
<?php
require_once "dbconnection.php";
require_once "session.php";
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {
    $fullname = $_POST['name'];
    $phn = $_POST['ph'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
    if($password==$confirm_password){
    $qry = "INSERT INTO `users` (`email`, `name`, `pnumber`) VALUES ('$email',
'$fullname', '$phn')";
    $run = mysqli_query($dbcon,$qry);
    if($run==true){
        $psqry = "INSERT INTO `login` (`email`, `password`, `u_id`) VALUES ('$email',
'$password',LAST_INSERT_ID() )";
        $psrun = mysqli_query($dbcon,$psqry);
        ?>  <script>
            alert('Registration Successfully :)');
            window.open('index.php','_self');
            </script>
        <?php
    }
    }else{
        ?>  <script>
            alert('Password mismatched!!');
            </script>
        <?php
    }
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Sign Up</title>
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
        <style>
        body
        {
        background-image:url('images/brr.png');
        background-repeat: no-repeat;
        background-size: cover;
        }
    </style>
    </head>
    <body><br>
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <h2 style="color:green">Register</h2>
                    <p>Please fill this form to create an account.</p>
                    <!-- <?php echo $success; ?>
                    <?php echo $error; ?> -->
                    <form action="" method="post">
                        <div class="form-group">
                            <label>Full Name</label>
                            <input type="text" name="name" class="form-control" required>
                        </div>
                        <div class="form-group">
                            <label>Phone Num.</label>
                            <input type="number" name="ph" class="form-control" required>
                        </div>
                        <div class="form-group">
                            <label>Email Address</label>
                            <input type="email" name="email" class="form-control" required />
                        </div>
                        <div class="form-group">
                            <label>Password</label>
                            <input type="password" name="password" class="form-control" required>
                        </div>
                        <div class="form-group">
                            <label>Confirm Password</label>
                            <input type="password" name="confirm_password" class="form-control"
required>
                        </div>
                        <div class="form-group">
                            <input type="submit" name="submit" class="btn btn-danger"
value="Register">
                        </div>
                        <p>Already have an account? <a href="index.php" style="color: red;">Login
here</a>.</p>
                    </form>
                </div>
            </div>
            <hr><p>Notice: If the email Id is registered before, it will not respond.</p>
            <p>In this case, reset your password or register with different email Id....</p>
        </div>
    </body>
</html>
```

ADMIN

Head

```
<!-- Admin header style for Admin pages -->
<!DOCTYPE html>
<html>
```

```
    <head>
            <title>Header Demo</title>
            <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,initial-scale=1.0">
  <style>
    *{
  margin: 0;
  padding: 0;
  background-repeat: no-repeat;
  }
  /* body {
  background-image: url('../images/plane.jpg');
  background-size: cover;
  } */
  .admintitle{
    position: relative;
    background-color: brown;
    color: #fff;
    height: 100px;
    line-height: 140px;
  }
  </style>
 </head>
    <body bgcolor="#067d64">
```

Admin login

```
<!-- admin logIn page and login logic -->
<?php


session_start();
if (isset($_SESSION['uid'])) {
  header('location: dashboard.php');
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Login</title>
</head>
<body bgcolor="#067d64">
    <h5><a href="../index.php" style="float: right; margin-right:50px;
color:#00BCD4">BackToHome</a></h5><br>
    <h1 align='center' style="color: #00BCD4;font-size:60px">Admin Login</h1>
    <h6 align='center' style="color: #212121;font-weight: bold;font-
size:15px">WELCOME</h6>
    <form action="adminlogin.php" method="POST" style="margin: auto;">
        <table align="center">
            <tr>
                <td>Email_ID:</td>
                <td><input type="email" name="uname" require></td>
            </tr>
            <tr><td><br></td></tr>
            <tr>
                <td>Password:</td>
                <td><input type="password" name="pass" require></td>
            </tr>
            <tr>
                <td colspan="2">
                    <hr>
                </td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input type="submit" name="login"
value="Login" style="cursor: pointer;"></td>
            </tr>
        </table>
    </form>
</body>
</html>



<?php
include('../dbconnection.php');
if (isset($_POST['login'])) {
    $ademail = $_POST['uname'];
    $password = $_POST['pass'];
    $qry = "SELECT * FROM `adlogin` WHERE `email`='$ademail' AND
`password`='$password'";
    $run = mysqli_query($dbcon, $qry);
    $row = mysqli_num_rows($run);
    if ($row < 1) {
        ?>
        <script>
            alert("Only admin can login..");
            window.open('adminlogin.php', '_self');
        </script><?php
    }
    else {
        $data = mysqli_fetch_assoc($run);
        $id = $data['a_id'];
        $_SESSION['uid'] = $id;
        header('location:dashboard.php');
    }
}
?>
```

USER SIDE

Header

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>Navbar with Logo Image</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
    .bs-example{
        margin: 0;
    }
</style>
</head>
<body>
<div class="bs-example">
    <nav class="navbar navbar-expand-md navbar-light bg-light">
        <a href="home.php" class="navbar-brand">
            <img src="../images/fcmw.png" height="80" alt="CoolBrand">
        </a>
        <button type="button" class="navbar-toggler" data-toggle="collapse" data-
target="#navbarCollapse">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
            <div class="navbar-nav">
                <a href="home.php" class="nav-item nav-link active">Home</a>
                <a href="price.php" class="nav-item nav-link active">Price</a>
                <a href="courierMenu.php" class="nav-item nav-link">Courier</a>
                <a href="trackMenu.php" class="nav-item nav-link">Track</a>
                <a href="profile.php" class="nav-item nav-link">Profile</a>
                <a href="contactUS.php" class="nav-item nav-link">ContactUs</a>
            <!-- mailto:premkumar1215225@gmail.com -->
          </div>
          <div class="navbar-nav ml-auto">
            <a href="../admin/logout.php" class="nav-item nav-link">AdminPage</a>
            <a href="../logout.php" class="nav-item nav-link">LogOut</a>
          </div>
        </div>
    </nav>
</div>
</body>
</html>
<?php include('footer.php'); ?>
```

Home

```
<?php
session_start();
if(isset($_SESSION['uid'])){
    echo "";
    }else{
    header('location: ../index.php');
    }
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home Page</title>
    <style>
        body
        {
        background-image:url('../images/1920_1080.jpg');
        background-repeat: no-repeat;
        background-size: cover;
        }
    </style>
</head>
<body>
    <?php include('header.php'); ?>
    <div align='center' style="font-weight: bold;font-family:'Times New Roman', Times,
serif"><br><br><br><br>
        <h2>This is a Typhoon Courier Management Service</h2>
        <h4>The fastest courier service of India</h4><br><br>
        <h3>DBMS MINI PROJECT</h3>
    </div>
</body>
</html>
```

Price

```
<?php
session_start();
if(isset($_SESSION['uid'])){
    echo "";
    }else{
    header('location: ../index.php');
    }
?>

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

<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pricing</title>
</head>
<body>
    <table width='30%' border="5px solid" style="margin-top:30px;margin-left:auto ;margin-
right:auto ;font-weight:bold;border-spacing: 5px 5px;border-collapse: collapse;">
    <tr style="background-color: green;font-size:30px">
    <th>Weight in Kg</th><th>Price</th>
    </tr>
    <tr>
    <td>0-1</td><td>120</td>
    </tr>
    <tr>
    <td>1-2</td><td>200</td>
    </tr>
    <tr>
    <td>2-4</td><td>250</td>
    </tr>
    <tr>
    <td>4-5</td><td>300</td>
    </tr>
    <tr>
    <td>5-7</td><td>400</td>
    </tr>
    <tr>
    <td>7-above</td><td>500</td>
    </tr>
    </table>
    <h3 align="center" style="margin-top:20px;"> As per your courier's weight pay the
amount on:</h3>
    <div style="margin-left:45% ;margin-right:auto ;font-weight:bold;">
    <ol>
    <li>UPI: 6360618467@ybl</li>
    <li>GPay: 6360618467</li>
    <li>PhnPay: 8340686169</li>
    </ol>
    </div>
</body>
</html>
```

Profile

```
<?php
session_start();
if(isset($_SESSION['uid'])){
    echo "";
    }else{
    header('location: ../index.php');
    }
?>
<?php
include('header.php');
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Profile</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
</head>
<body>
<?php
```

```
include('../dbconnection.php');
$id= $_SESSION['uid'];
$qry= "SELECT * FROM `users` WHERE `u_id`='$id'";
$run= mysqli_query($dbcon,$qry);
$data = mysqli_fetch_assoc($run);
?>
<div class="page-content page-container" id="page-content">
    <div class="padding">
        <div class="row container d-flex justify-content-center">
            <div class="col-xl-6 col-md-12">
                <div class="card user-card-full">
                    <div class="row m-l-0 m-r-0">
                        <div class="col-sm-4 bg-c-lite-green user-profile">
                            <div class="card-block text-center text-white">
                                <div class="m-b-25"> <img
src="https://img.icons8.com/bubbles/100/000000/user.png" class="img-radius" alt="User-
Profile-Image"> </div>
                                <h3 class="f-w-600"><?php echo $data['name']; ?></h3>
                                <p>user</p> <i class=" mdi mdi-square-edit-outline feather icon-edit m-
t-10 f-16"></i>
                            </div>
                        </div>
                        <div class="col-sm-8">
                            <div class="card-block">
                                <h6 class="m-b-20 p-b-5 b-b-default f-w-600">Information</h6>
                                <div class="row">
                                    <div class="col-sm-6">
                                        <p class="m-b-10 f-w-600">Email</p>
                                        <h6 class="text-muted f-w-400"><?php echo $data['email'];
?></h6>
                                    </div>
                                    <div class="col-sm-6">
                                        <p class="m-b-10 f-w-600">Phone</p>
                                        <h6 class="text-muted f-w-400"><?php echo $data['pnumber'];
?></h6>
                                    </div>
                                </div>
                               div class="row">
                                <hr><br><hr>
                                </div>
                                <ul class="social-link list-unstyled m-t-40 m-b-10">
                                <h6>Leave it when u can't hold it..</h6>
                                </ul>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</body>
</html>
```

CHAPTER 5

RESULTS, SNAPSHOTS AND DISCUSSIONS

HOMEPAGE: Fig.5.1. Represents the Homepage of our website

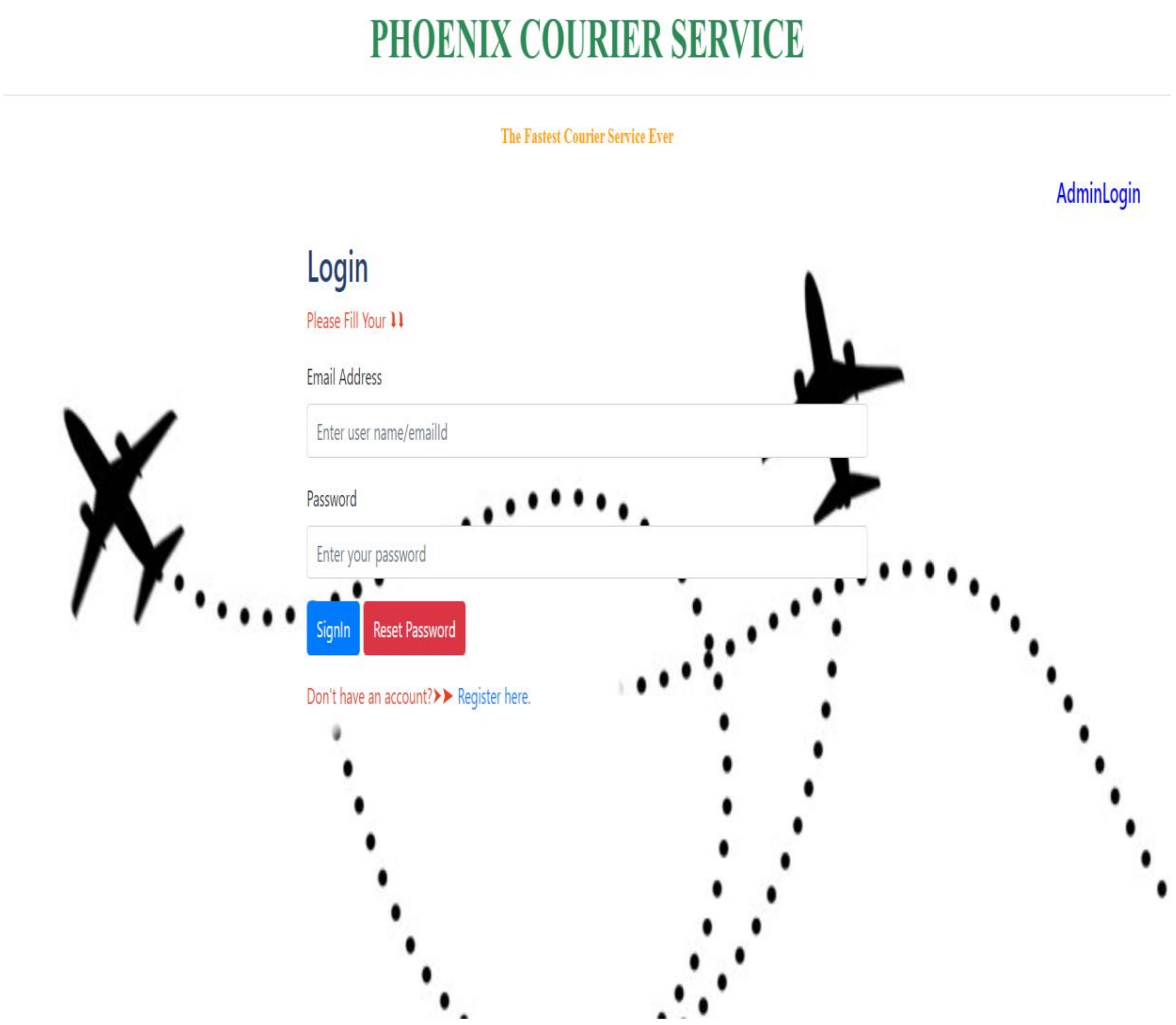
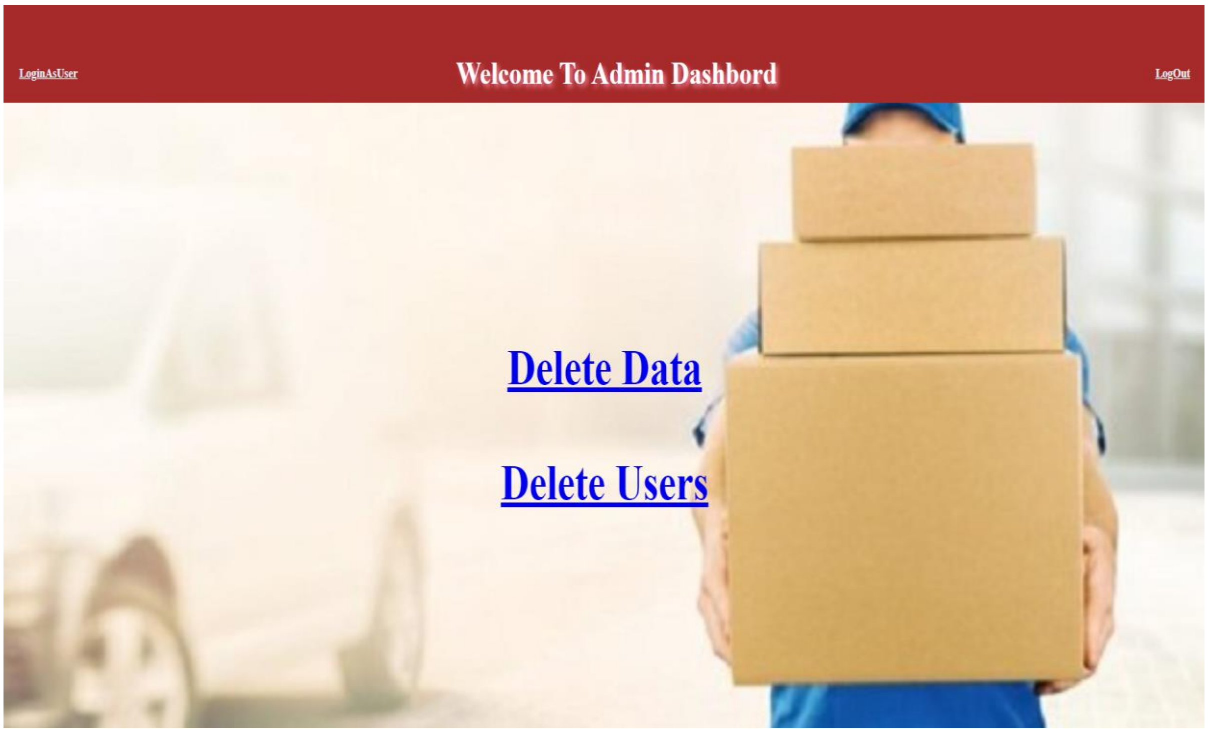
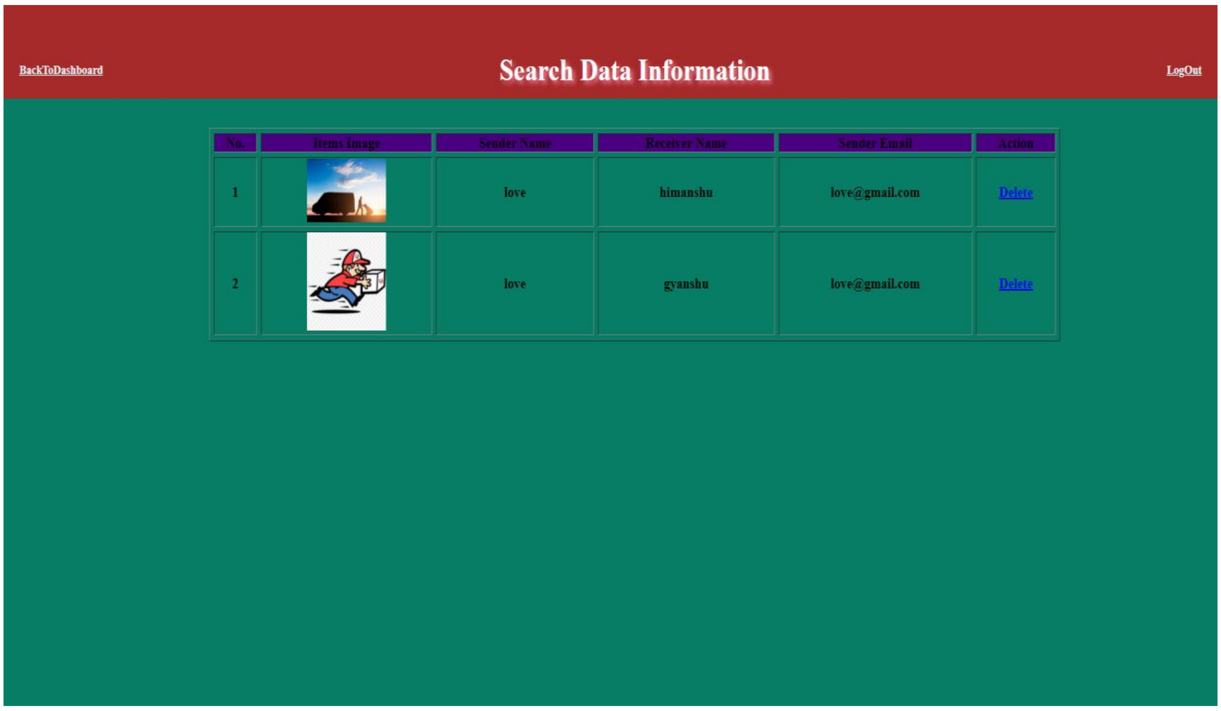


Fig 5.1 HOMEPAGE

ADMIN DASHBOARD: Fig 5.2 Admin Can Manage and View Data .

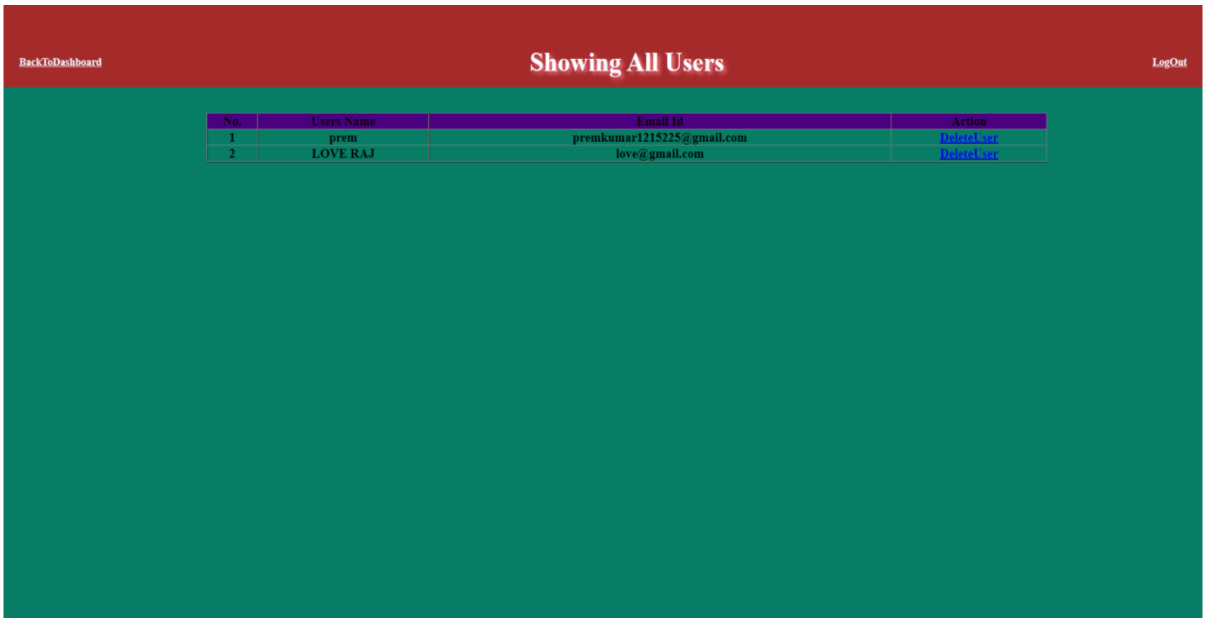




No.	Items Image	Sender Name	Receiver Name	Sender Email	Action
1		love	himanshu	love@gmail.com	Delete
2		love	gyanshu	love@gmail.com	Delete

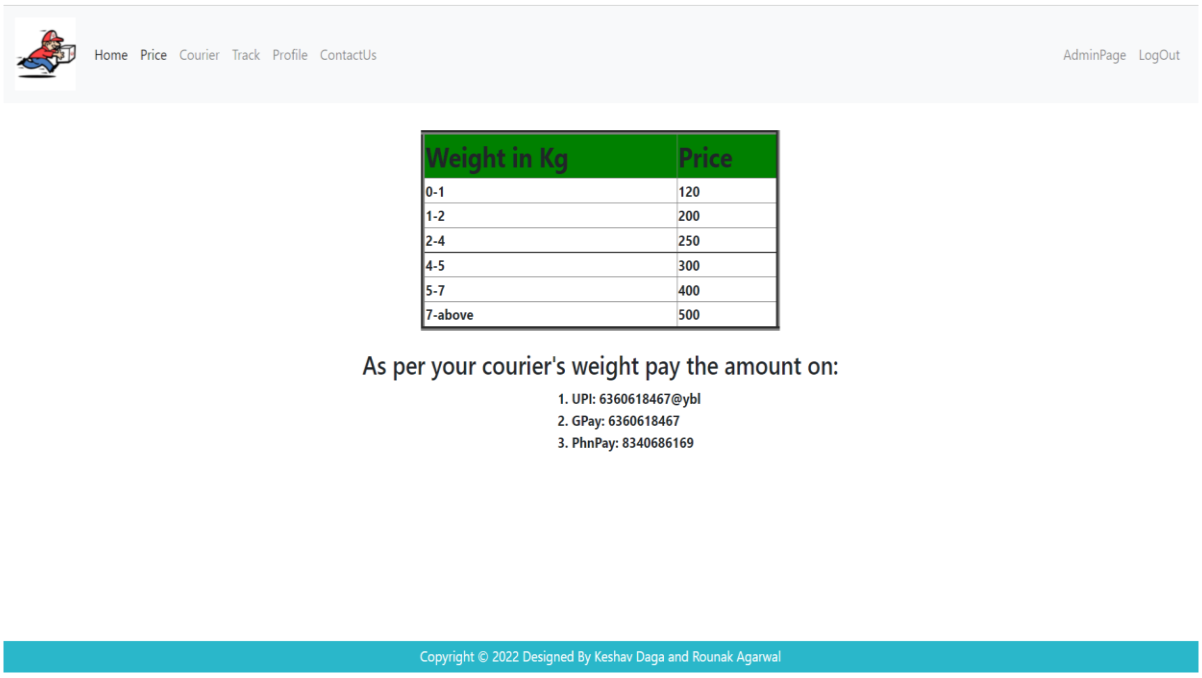
Fig 5.2 Admin Dashboard

ADMIN ACCESSING DETAILS: Fig 5.3 Admin can view the parcel orders underway and current users.



Showing All Users

No.	Users Name	Email Id	Action
1	prem	premkumar1215225@gmail.com	DeleteUser
2	LOVE RAJ	love@gmail.com	DeleteUser



Weight in Kg	Price
0-1	120
1-2	200
2-4	250
4-5	300
5-7	400
7-above	500

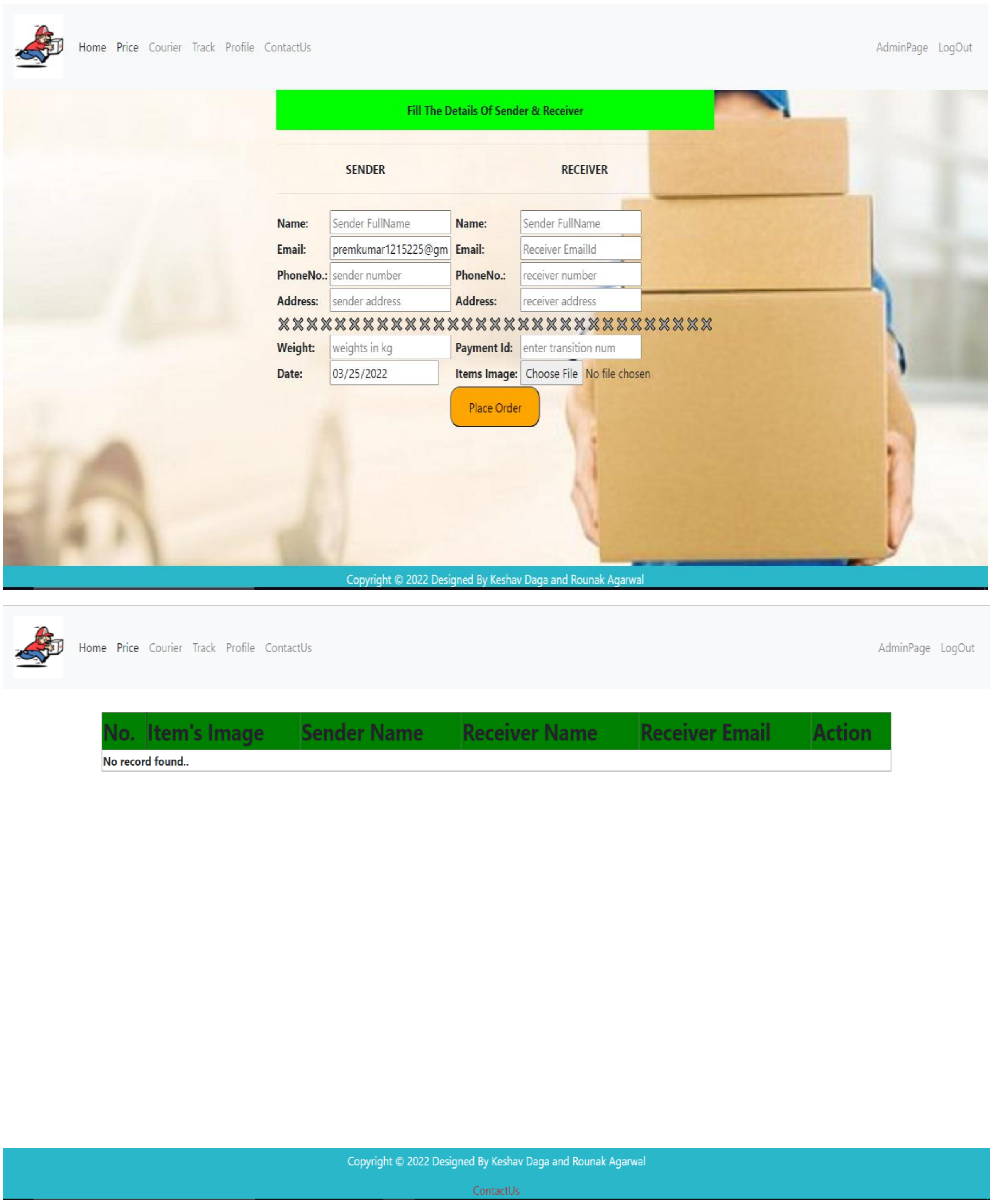
As per your courier's weight pay the amount on:

1. UPI: 6360618467@ybl
2. GPay: 6360618467
3. PhnPay: 8340686169

Copyright © 2022 Designed By Keshav Daga and Rounak Agarwal

Fig 5.3 ADMING ACCESING DETAILS

NEW PARCEL ORDER: Fig 5.4 New Order details can be entered.



Fill The Details Of Sender & Receiver

SENDER		RECEIVER	
Name:	Sender FullName	Name:	Sender FullName
Email:	premkumar1215225@gm	Email:	Receiver EmailId
PhoneNo.:	sender number	PhoneNo.:	receiver number
Address:	sender address	Address:	receiver address
Weight:	weights in kg	Payment Id:	enter transition num
Date:	03/25/2022	Items Image:	Choose File No file chosen

Place Order

Copyright © 2022 Designed By Keshav Daga and Rounak Agarwal

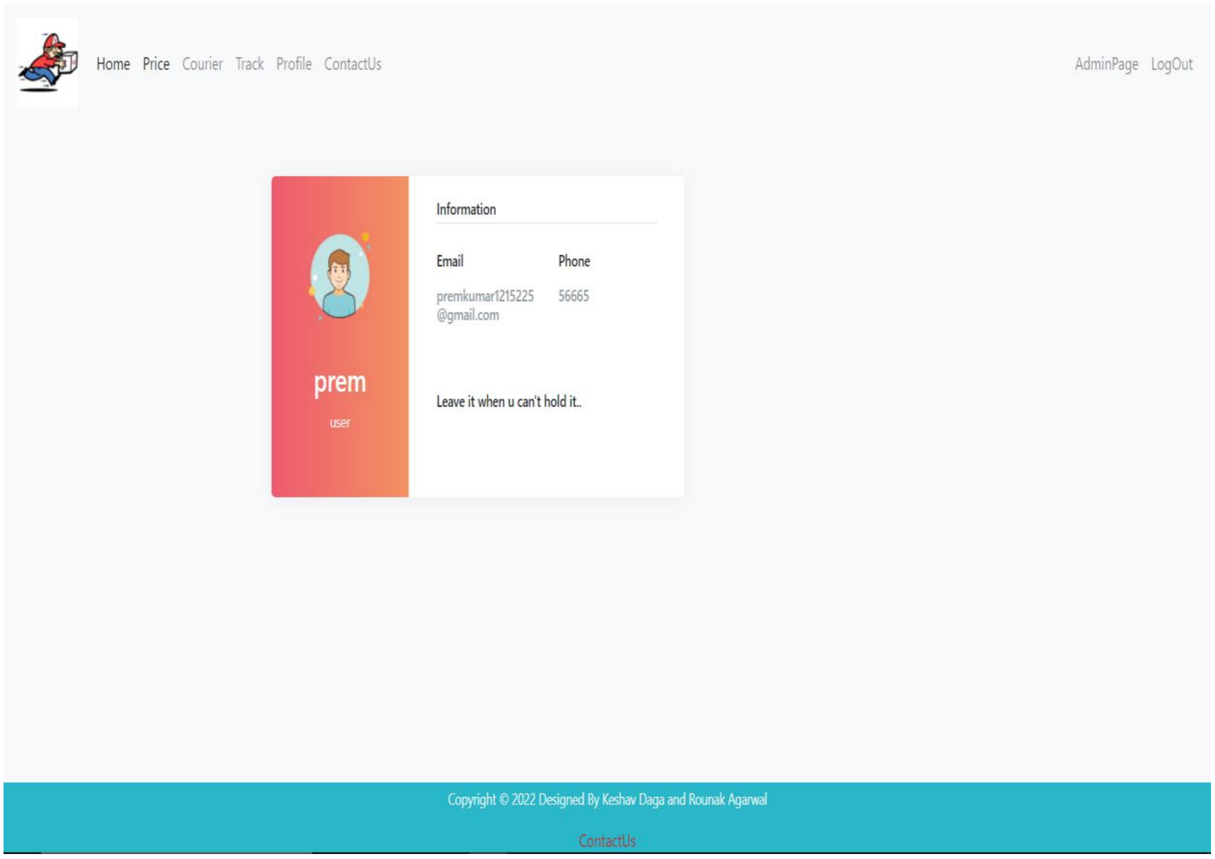
No.	Item's Image	Sender Name	Receiver Name	Receiver Email	Action
No record found..					

Copyright © 2022 Designed By Keshav Daga and Rounak Agarwal

ContactUs

Fig 5.4 NEW PARCEL ORDER

USER PROFILE: Fig 5.5 It shows User details .



prem

user

Information

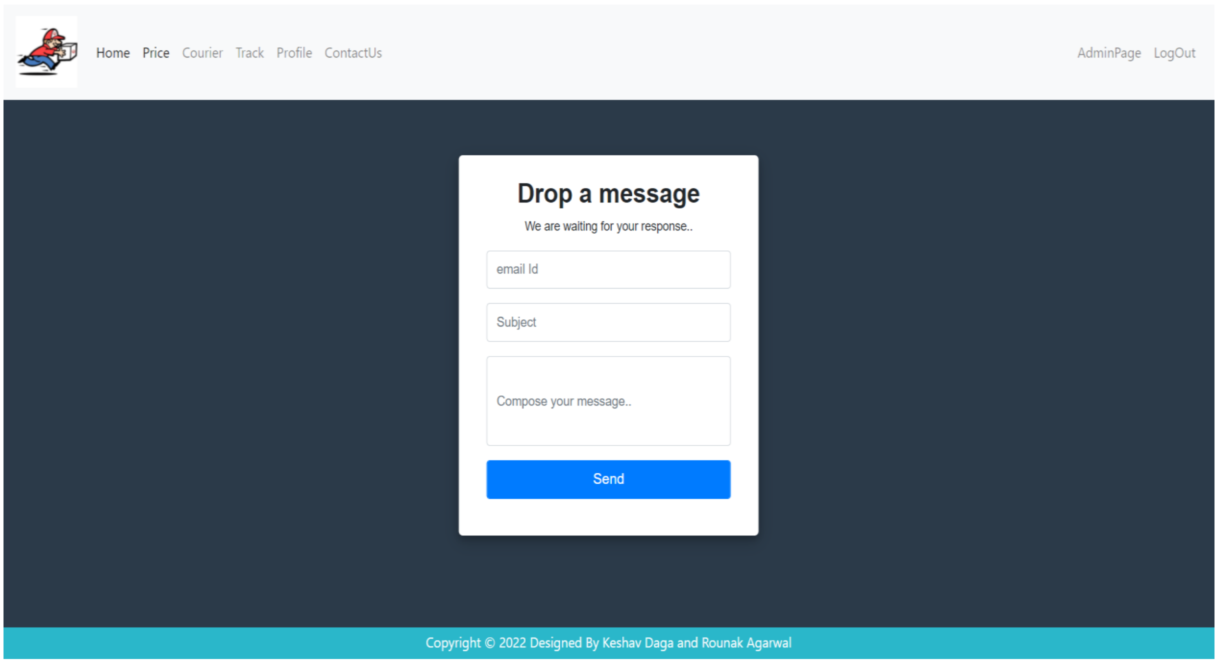
Email	Phone
premkumar1215225@gmail.com	56665

Leave it when u can't hold it..

Copyright © 2022 Designed By Keshav Daga and Rounak Agarwal

Fig 5.5 USER PROFILE

CONTACT INTERFACE: Fig 5.6 User can raise grievances to admin.



Drop a message

We are waiting for your response..

email Id

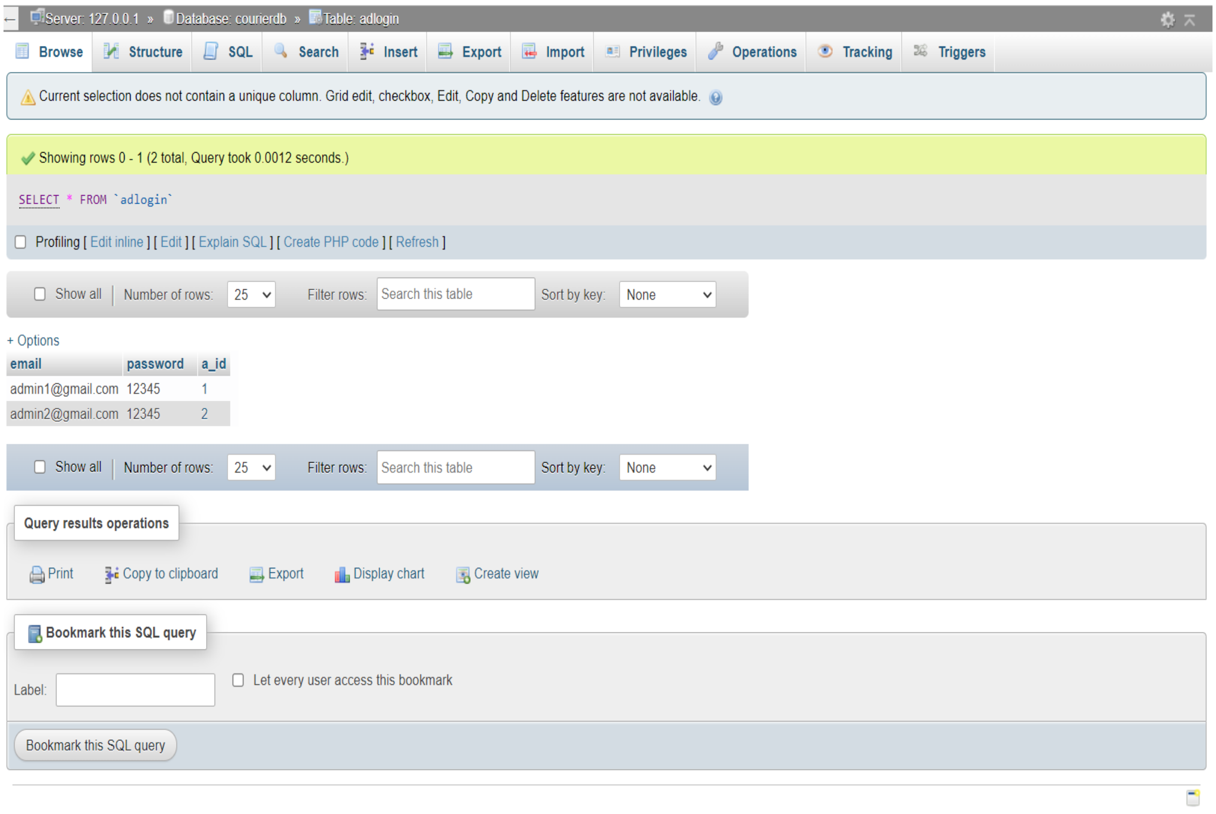
Subject

Compose your message..

Copyright © 2022 Designed By Keshav Daga and Rounak Agarwal

Fig 5.6 CONTACT INTERFACE

COURIER DATABASE: Fig 5.7 It contains information and data stored in DBMS.



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0012 seconds.)

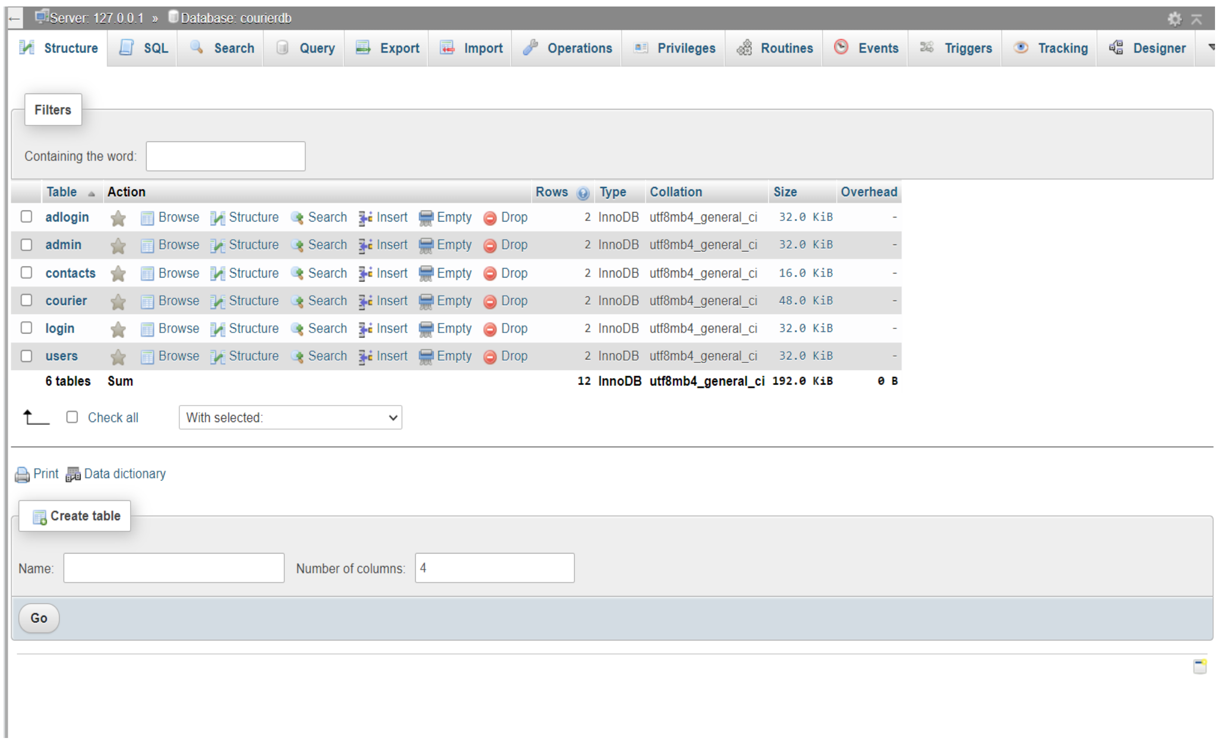
```
SELECT * FROM `adlogin`
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

email	password	a_id
admin1@gmail.com	12345	1
admin2@gmail.com	12345	2

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

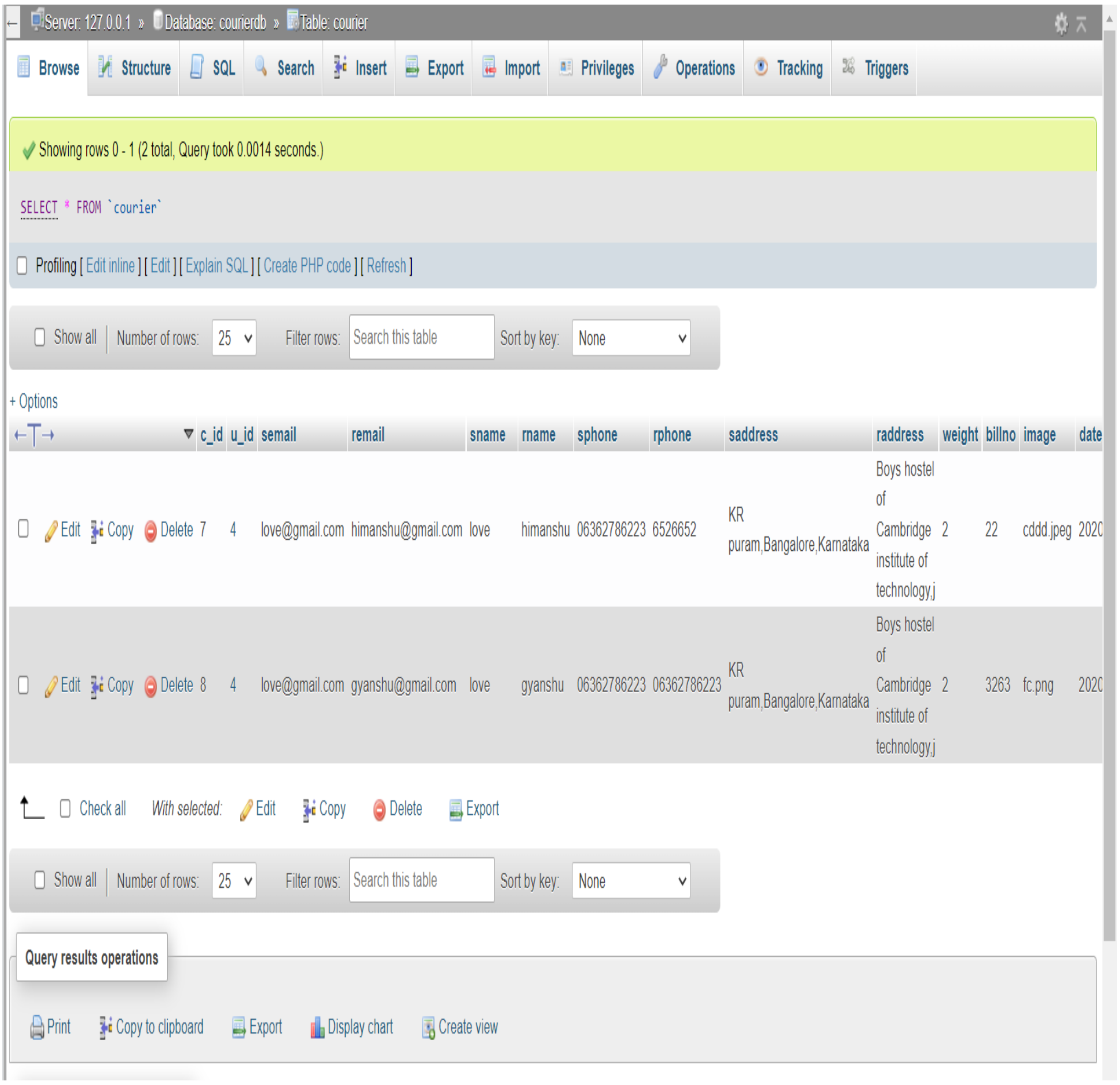
Label:



Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
adlogin	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
admin	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
contacts	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
courier	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	48.0 KiB	-
login	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
6 tables	Sum	12	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B

Name: | Number of columns: 4


Server: 127.0.0.1 » Database: courierdb » Table: courier
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

Showing rows 0 - 1 (2 total, Query took 0.0014 seconds.)

```
SELECT * FROM `courier`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options

	c_id	u_id	semail	remail	sname	rname	sphone	rphone	saddress	raddress	weight	billno	image	date
Edit Copy Delete	7	4	love@gmail.com	himanshu@gmail.com	love	himanshu	06362786223	6526652	KR puram,Bangalore,Karnataka	Boys hostel of Cambridge institute of technology,j	2	22	cddd.jpeg	2020
Edit Copy Delete	8	4	love@gmail.com	gyanshu@gmail.com	love	gyanshu	06362786223	06362786223	KR puram,Bangalore,Karnataka	Boys hostel of Cambridge institute of technology,j	2	3263	fc.png	2020

Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view

Fig 5.7 COURIER DATABASE

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

System development is also considered as a process backed by engineering approach. We have tried to incorporate & develop new particles for our education particles have been followed not during the but coding but also during the analysis, design phases & in documentation.

Courier agency is considered as an expansion of business relations. It contributes a lot by providing quick & fast services of sending documents letters (formal & informal both) to business as it enables any business to flourish

6.2FUTURE ENHANCEMENTS

Following modification or upgrades can be done in system.

1) More than one company can be integrated through this software.
2) Web services can be used to know exact delivery status of packets.
3) Client can check the repacked delivery status online.
4) Distributed database approach in place of centralized approach

BIBLIOGRAPHY

1. Ramez Elmarsi and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.
2. http://www.bluedart.com/
3. http://www.xamppserver.com/en/
4. http://www.php.net/
5. http://youtube.com/
6. http://www.tutorialspoint.com/mysql/
7. https//apache.org/docs/2.0/misc/tutorials.html