

CHAPTER 1

INTRODUCTION

1.1 Introduction To The Project

This Courier Management System Project will have different modules. The login section will have a log in facility for the admin and for the user who will operate this system. While taking orders from its customers, it will take all the details of its customers who are placing the orders and all the details for the recipient such as their address, name, and, mobile number. During the billing

process system will generate a tracking id for their products. Through this tracking id, customers or its recipient will able to track their products from any location using the internet. It

will provide the status of the product after placing orders within 1 minute.

The courier service is one of the solutions of these problems. It is used to send some things to any person in the world within time. The courier company has number of branches, which are spread over the country or the world. So that when person wants to send things then he has to contact at nearest courier service branch. The courier company creates the schedule & gives internal/external services. The courier service work as destination office or source office.

In modern age, as time increase, needs & requirements of the person are also increased. They want more facility & try to do their task quickly & within time. But they cannot get all the things at nearest market or area, so they have to import the things from any place in the world. Within the country, the things can be imported through post service. But it consumes the time & sometimes problem of damage or missing occur. Where as in the international market, the one way is shipping. But it also requires more time.

1.2 OBJECTIVE:

To gain maximum business region, customer demands good service. So to make more profits

and gain maximum business region, their administration must also have a system to tackle all these problems on time. Its administration can take immediate orders and provide a receipt which will include all the details of the products along with appropriate price to their customers. Thus saving time and eliminating line making process.

1.3 Features of Purposed System:

These are the important features of the project Courier Management System:

- In computer system of the courier service computation of the rate is easily & quickly done.
- Computer system of the courier service provide fast access.
- Using this computerized system, bill issued procedure becomes fast.
- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the Manifest but we can directly print it, which saves our time.
- It contain better storage capacity.
- Accuracy in work.
- Easy & fast retrieval of information.
- Well designed reports.
- Decrease the load of the person involve in existing manual system.
- Access of any information individually.
- Work becomes very speedy and easy to update information

1.4 Traditional File System

File System is collection of data. In this system, user has to write procedures for managing database. It provides details of data representation and storage of data. In this –

- Data is stored in files.
- Each file has specific format.
- Programs that use these files depend on knowledge about that format.
- In earlier days, database applications were built on top of file systems.

Basically, it is a collection of application programs that performs services for end users such as production of reports. Each file defines and manages its own data.

1.4.1 Pros And Cons Of Traditional Approach

Pros:

- File Processing cost less and can be faster than Database.
- File Processing design approach was well suited to mainframe hardware and batch input.
- Companies mainly use file processing to handle large volumes of structured data on a regular basis.
- It can be more efficient and cost effective in many cases.
- Design is simple.
- Customization is easy.

Cons:

- Data Redundancy and Inconsistency.
- Difficulty in accessing data.
- Data Isolation – Multiple files and formats.
- Integrity problems
- Unauthorized Access is not restricted.
- It Co-Ordinates only physical access.

1.5 Introduction To DBMS

Databases and database technology have had a major impact on the growing use of computers. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. Nowadays, this data is typically stored in mobile phones, which have their own simple database software. In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. A database can be of any size and complexity. For example, the list of names and addresses referred to earlier may consist of only a few hundred records, each with a simple structure. On the other hand, the computerized catalogue of a large library may contain half a million entries organized under different categories.

A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalogue or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

1.5.1 Advantages Of DBMS

Compared to the File Based Data Management System, Database Management System has many advantages.

1. Reducing Data Redundancy

The file-based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which led to data redundancy.

This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

2.Data Integrity

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data

that is visible to multiple users. So, it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

3.Data Security

Data Security is vital concept in a database. Only authorized users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorized users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

4.Privacy

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

5.Backup and Recovery

Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

6.Data Consistency

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

CHAPTER 2

SYSTEM REQUIREMENTS

This section describes the software and hardware requirements of the system.

2.1 - Software Requirement:

- Apache Server 2.0
- PHP Version 5.3 or above
- MySQL Version 5.5 or above
- Latest browser: Chrome, Firefox, Safari etc
- Operating System: Any (Linux, Windows, Mac etc)

2.2- Hardware Requirements:

- Processor Pentium IV or higher version.
- Ram 128 MB or above
- Hard Disk 150 MB or above

CHAPTER 3

DATABASE DESIGN

3.1-Requirements And Constraints

3.1.1 Functional Requirements:

1-Distributed Database:

Distributed Database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a Communication Network.

2-Client/Server System

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

3-User Interfaces

Front-End Software: HTML, CSS, JAVASCRIPT, BOOTSTRAP.

Back-End Software: MySQL

4-Hardware Interfaces

- Windows.
- A browser which supports php and html

5-Security Requirements

- Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

3.1.2 Constraints

Constraints on The Relational Database Are Of 4 Types:

1. Domain Constraints
2. Key Constraints
3. Entity Integrity Constraints
4. Referential Integrity Constraints

1. Domain Constraints:

1. Every domain must contain atomic values (smallest indivisible units) it means composite and multi-valued attributes are not allowed.
2. We perform datatype check here, which means when we assign a data type to a column, we limit the values that it can contain.
E.g., If we assign the datatype of attribute age as int, we can't give it values other than int datatype.

2. Key Constraints Or Uniqueness Constraints:

1. These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.
2. A relation can have multiple keys or candidate keys(minimal super key), out of which we choose one of the keys as primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with less number of attributes.
3. Null values are not allowed in the primary key, hence not null constraint is also a part of key constraint.

3. Entity Integrity Constraints:

1. Entity integrity constraints says that no primary key can take null value, since using primary key we identify each tuple uniquely in a relation.

4. Referential Integrity Constraints:

1. The referential integrity constraints is specified between two relations or — tables and used to maintain the consistency among the tuples in two relations.
2. This constraint is enforced through foreign key, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2,

then the foreign key of r1 is said to reference or refer to the primary key of relation R2.

3. The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take null values, but can't be empty.

3.2 Entities and Attribute

1) ADLOGIN

- i) Email
- ii) Password
- iii) A_Id

2) ADMIN

- i) A_Id
- ii) Email
- iii) Name
- iv) Pnumber

3) CONTACTS

- i) Id
- ii) Email
- iii) Subject
- iv) Msg

4) COURIER

- i) C_Id
- ii) U_Id
- iii) Semail
- iv) Remail
- v) Sname
- vi) Rname
- vii) Sphone
- viii) Rphone
- xi) Saddress
- x) Raddress
- xi) Weight
- xii) Billno
- xiii) Image
- xiv) Date

5) LOGIN

- i) Email
- ii) Password
- iii) U_Id

6) LOGSS

- i) User_Id
- ii) Action_Time
- iii) Action_Performed
- iv) Action_Performed_By

7) USERS

- i) U_Id
- ii) Email
- iii) Name
- iv) Pnumber

3.3 ENTITY RELATIONSHIP DIAGRAM

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

What is an Entity Relationship Diagram (ER Diagram)?

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

The geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section (Components of a ER Diagram) of this guide so don't worry too much about these terms now, just go through them once.

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double Lines: Total participation of an entity in a relationship set

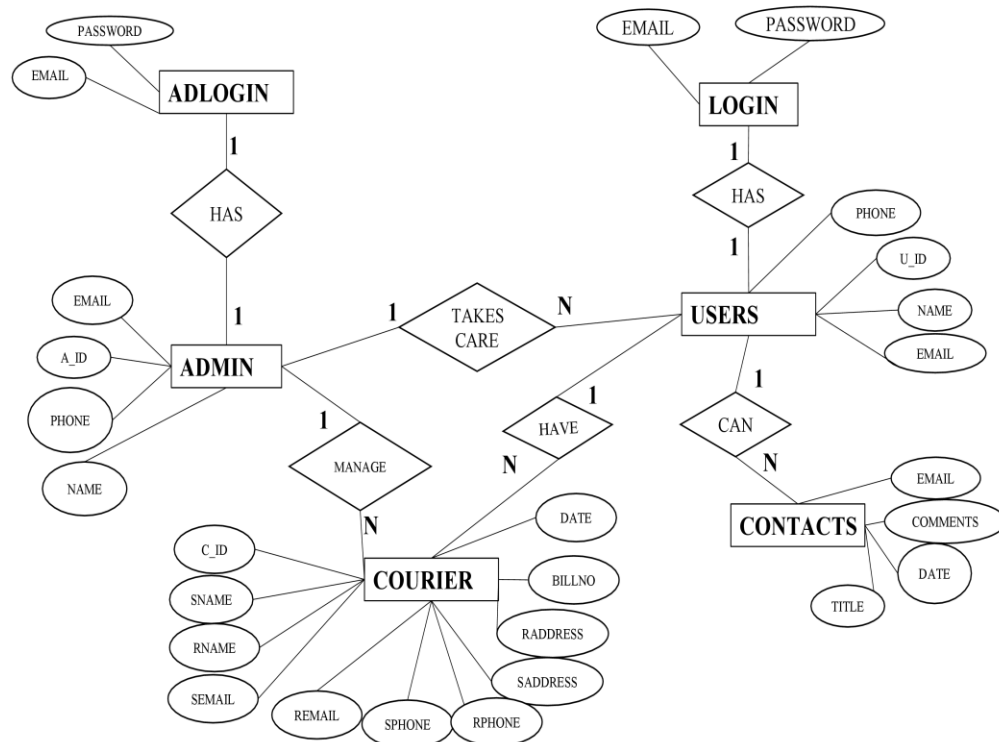


Fig 3.1 ER Diagram For Courier Management System

In the ER diagram, we have

1. **Entity set:** adlogin, login, admin, user, courier, contacts.
2. **Attribute:** (password, email),(email, password),(email,a_id,phone,name),(c_id,sname,rname,semail,remail,sphone,rphone,saddress,raddress,billno,date),(email,comments,date,title)
3. **Relationship:** has, takes care, manage, have, can.

3.4 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

An Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

In order to create an ER schema you must know three main concepts: entity, attribute and relationship.

Entity

The entity is the central concept of the Entity-Relationship model. An entity represents a description of the common features of the set of objects in the real world. Examples of entities are Person, Car, Artist, and Album.

Attribute

An Attribute represents the properties of real world objects that are relevant for the application purposes. Attributes are associated with the concept of Entity, with the meaning that all the instances of the entity are characterized by the same set of attributes. In other words, the entity is a descriptor of the common properties of a set of objects, and such properties are expressed as attributes.

Relationship

A Relationship represents semantic connections between entities, like the association between an artist and his/her album, or between an artist and his/her reviews.

The possible values are one and many. Based on their maximum cardinality constraints, relationships are called

1. "one-to-one", if both relationships roles have maximum cardinality 1,
2. "one-to-many", if one relationship role has maximum cardinality 1 and the other role has maximum cardinality N,
3. "many-to-many", if both relationships roles have maximum cardinality N.

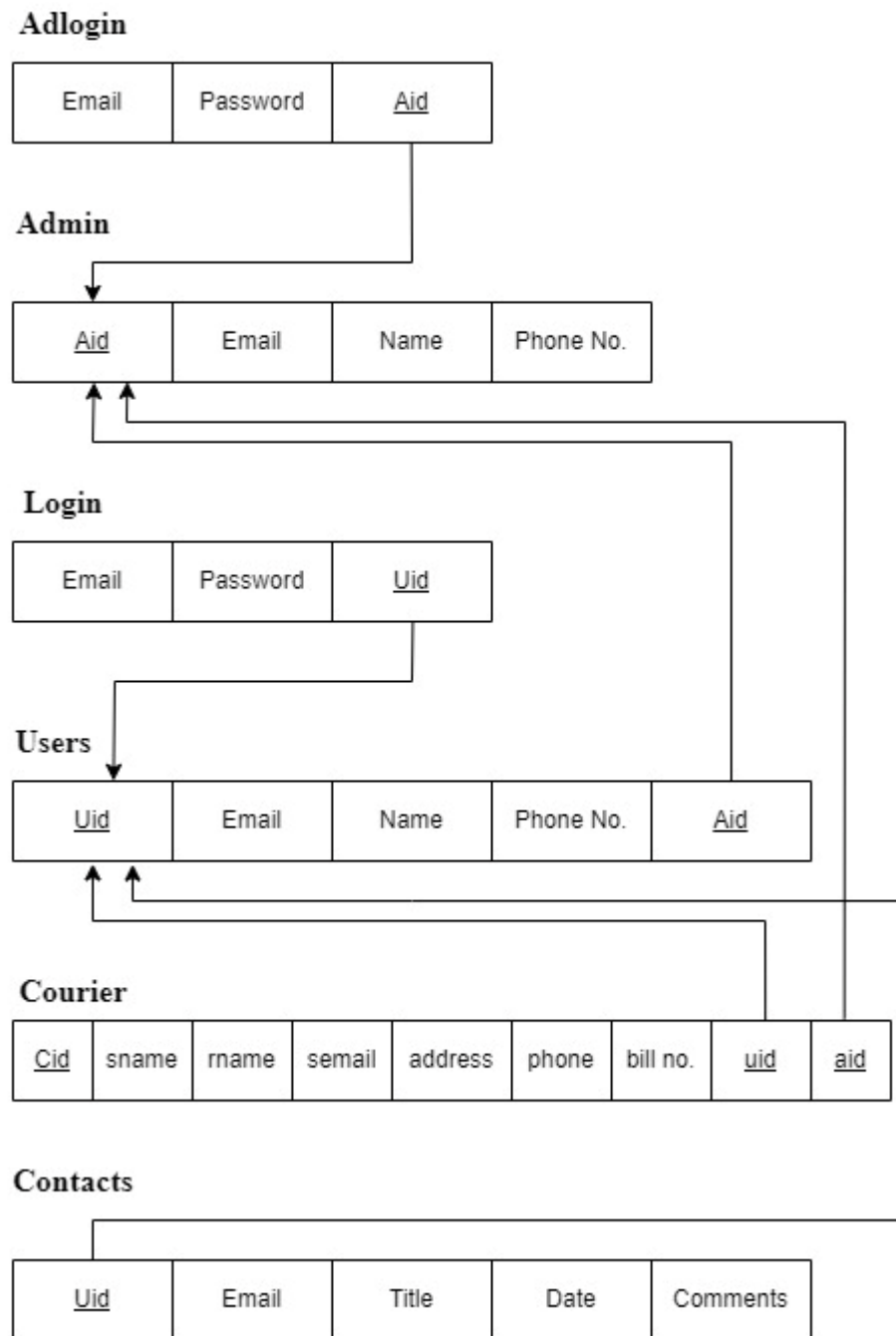


Fig 3.2 Schema Diagram For Courier Management System

CHAPTER 4

IMPLEMENTATION

4.1 Frontend Code

4.1.1 INDEX.PHP

```
<?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! Please Enter Your Username and Password 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 🙌");
            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">Keerthana & Krishna 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 type="button" onclick="window.location='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>
</div>

```

```
</body>
</html>
```

4.1.2 ADMINLOGIN.PHP

```
<!-- 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">
    <h3><a href="../index.php" style="float: right; margin-right:50px;
color:#00BCD4">BackToHome</a></h3><br>
    <h1 align='center' style="color: #00BCD4;font-size:60px">Admin Login</h1>
    <h2 align='center' style="color: #212121;font-weight: bold;font-size:30 px">welcome
Admin</h2>
    <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');
    }
}
?>

```

4.1.3 HOME.PHP

```

<?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/abc.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><b>Welcome To Keerthana & Krishna Courier Service</b></h2>
    <h3><b>#1 courier service in Asia</b></h3><br><br>
    <h4><b> DBMS MINI PROJECT</b></h4>
    <h6></h6>
  </div>
</body>
</html>

```

4.2 DBCONNECTION.PHP

```

<?php
$dbcon = mysqli_connect('localhost','root','','courierdb_sql');
if($dbcon==false)
{
  echo "Database is not Connected!";
}
?>

```

4.3 BACKEND CODE [COURIERDB_SQL]

```

-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Dec 17, 2022 at 01:58 AM
-- Server version: 10.4.22-MariaDB
-- PHP Version: 7.4.27
SET time_zone = "+00:00";

-- Database: `courierdb_sql`

-----
-- Table structure for table `adlogin`

CREATE TABLE `adlogin` (
  `email` varchar(50) DEFAULT NULL,
  `password` varchar(50) DEFAULT NULL,
  `a_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

-- Dumping data for table `adlogin`

```
INSERT INTO `adlogin` (`email`, `password`, `a_id`) VALUES
('keerthana@gmail.com', '12345', 1),
('krishna@gmail.com', '12345', 2);
```

-- Table structure for table `admin`

```
CREATE TABLE `admin` (
  `a_id` int(11) NOT NULL,
  `email` varchar(50) NOT NULL,
  `name` varchar(50) DEFAULT NULL,
  `pnumber` int(14) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Dumping data for table `admin`

```
INSERT INTO `admin` (`a_id`, `email`, `name`, `pnumber`) VALUES
(1, 'keerthana@gmail.com', 'Keerthana', 12345),
(2, 'krishna@gmail.com', 'Krishna', 12345);
```

-- Table structure for table `contacts`

```
CREATE TABLE `contacts` (
  `id` int(11) NOT NULL,
  `email` varchar(50) NOT NULL,
  `subject` varchar(30) NOT NULL,
  `msg` varchar(300) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Dumping data for table `contacts`

```
INSERT INTO `contacts` (`id`, `email`, `subject`, `msg`) VALUES
(5, 'keerthana@gmail.com', 'WRONG PACKAGE', 'I received a wrong package'),
(103, 'krishna@gmail.com', 'BROKEN ITEM', 'My very expensive ivory mug is broken, refund me');
```

-- Table structure for table `courier`

```
CREATE TABLE `courier` (
  `c_id` int(11) NOT NULL,
  `u_id` int(11) DEFAULT NULL,
  `semail` varchar(50) DEFAULT NULL,
  `remai` varchar(50) DEFAULT NULL,
```

```
`sname` varchar(50) DEFAULT NULL,
`rname` varchar(50) DEFAULT NULL,
`sphone` varchar(20) DEFAULT NULL,
`rphone` varchar(20) DEFAULT NULL,
`saddress` varchar(50) DEFAULT NULL,
`raddress` varchar(50) DEFAULT NULL,
`weight` int(11) DEFAULT NULL,
`billno` int(11) NOT NULL,
`image` text DEFAULT NULL,
`date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Dumping data for table `courier`

```
INSERT INTO `courier` (`c_id`, `u_id`, `semail`, `remail`, `sname`, `rname`, `sphone`,
`rphone`, `saddress`, `raddress`, `weight`, `billno`, `image`, `date`) VALUES
(13, 12, 'susan@gmail.com', 'krishna@gmail.com', 'susan', 'krishna', '123456', '654321', 'goa',
'noida', 5, 12, 'cddd.jpeg', '2022-12-18'),
(16, 13, 'keerthana@gmail.com', 'susan@gmail.com', 'keerthana', 'susan', '987654321',
'987123456', 'delhi', 'pune', 5, 11, '2959860-science-fiction-artwork-robot___abstract-
wallpapers.jpg', '2022-12-21'),
(17, 105, 'krishna@gmail.com', 'susan@gmail.com', 'krishna', 'susan', '2147483647',
'2474851756', 'noida', 'indore', 4, 10, 'fc.png', '2022-12-21');
```

-- Table structure for table `login`

```
CREATE TABLE `login` (
  `email` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `u_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Dumping data for table `login`

```
INSERT INTO `login` (`email`, `password`, `u_id`) VALUES
('susan@gmail.com', 'susan', 12),
('keerthana@gmail.com', 'keerthana', 13),
('krishna@gmail.com', 'krishna', 105);
```

-- Table structure for table `logss`

```
CREATE TABLE `logss` (
  `user_id` int(11) NOT NULL,
  `action_time` datetime NOT NULL,
  `action_performed` text NOT NULL,
  `action_performed_by` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Dumping data for table `logss`

```
INSERT INTO `logss` (`user_id`, `action_time`, `action_performed`, `action_performed_by`)
VALUES
(104, '2022-12-19 01:09:29', 'USER DELETED', 'USER'),
(1, '2022-12-19 01:09:43', 'USER DELETED', 'USER'),
(105, '2022-12-21 18:02:07', 'USER CREATED', 'USER'),
(102, '2022-12-21 18:11:57', 'USER DELETED', 'USER'),
(14, '2022-12-21 18:13:06', 'USER CREATED', 'USER');
```

-- Table structure for table `users`

```
CREATE TABLE `users` (
  `u_id` int(11) NOT NULL,
  `email` varchar(50) NOT NULL,
  `name` varchar(50) NOT NULL,
  `pnumber` int(14) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Dumping data for table `users`

```
INSERT INTO `users` (`u_id`, `email`, `name`, `pnumber`) VALUES
(12, 'susan@gmail.com', 'susan', 123456789),
(13, 'keerthana@gmail.com', 'keerthana', 987654321),
(14, 'lizzie@gmail.com', 'lizzie', 2118529630),
(105, 'krishna@gmail.com', 'krishna', 2147483647);
```

-- Triggers `users`

```
DELIMITER $$
```

```
CREATE TRIGGER `Customer_insert` AFTER INSERT ON `users` FOR EACH ROW
INSERT INTO Logss VALUES(NEW.u_id,NOW(),'USER CREATED','USER')
```

```
$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE TRIGGER `user_deleted` BEFORE DELETE ON `users` FOR EACH ROW
INSERT INTO Logss VALUES(OLD.u_id,NOW(),'USER DELETED','USER')
```

```
$$
```

```
DELIMITER ;
```

-- Constraints for dumped tables

-- Constraints for table `adlogin`

```
ALTER TABLE `adlogin`
  ADD CONSTRAINT `adlogin_ibfk_1` FOREIGN KEY (`a_id`) REFERENCES `admin`
  (`a_id`);
```

```
-- Constraints for table `courier`
ALTER TABLE `courier`
  ADD CONSTRAINT `courier_ibfk_1` FOREIGN KEY (`u_id`) REFERENCES `users`
  (`u_id`) ON DELETE CASCADE;
-- Constraints for table `login`
ALTER TABLE `login`
  ADD CONSTRAINT `login_ibfk_1` FOREIGN KEY (`u_id`) REFERENCES `users`
  (`u_id`) ON DELETE CASCADE;
COMMIT;
```

END OF IMPLEMENTATION CODE OF THE PROJECT

CHAPTER 5

TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing the program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is a black-box testing. System testing is performed after the integration testing and before the acceptance testing.

CHAPTER 6

SNAPSHOTS

Keerthana & Krishna Courier Services

#1 COURIER SERVICE IN ASIA

[Admin Login](#)

LOGIN

Please Fill Your Details Down Here

Email Address

Password

[Sign In](#)

[Don't have an account? Register here](#)

Fig 6.1 User Login Page

The fig 6.1 shows the User login page of Keerthana & Krishna Courier Service, the login page has the space to enter the username and the password if the user is already registered or if the user is new, there is a option to create a new account.

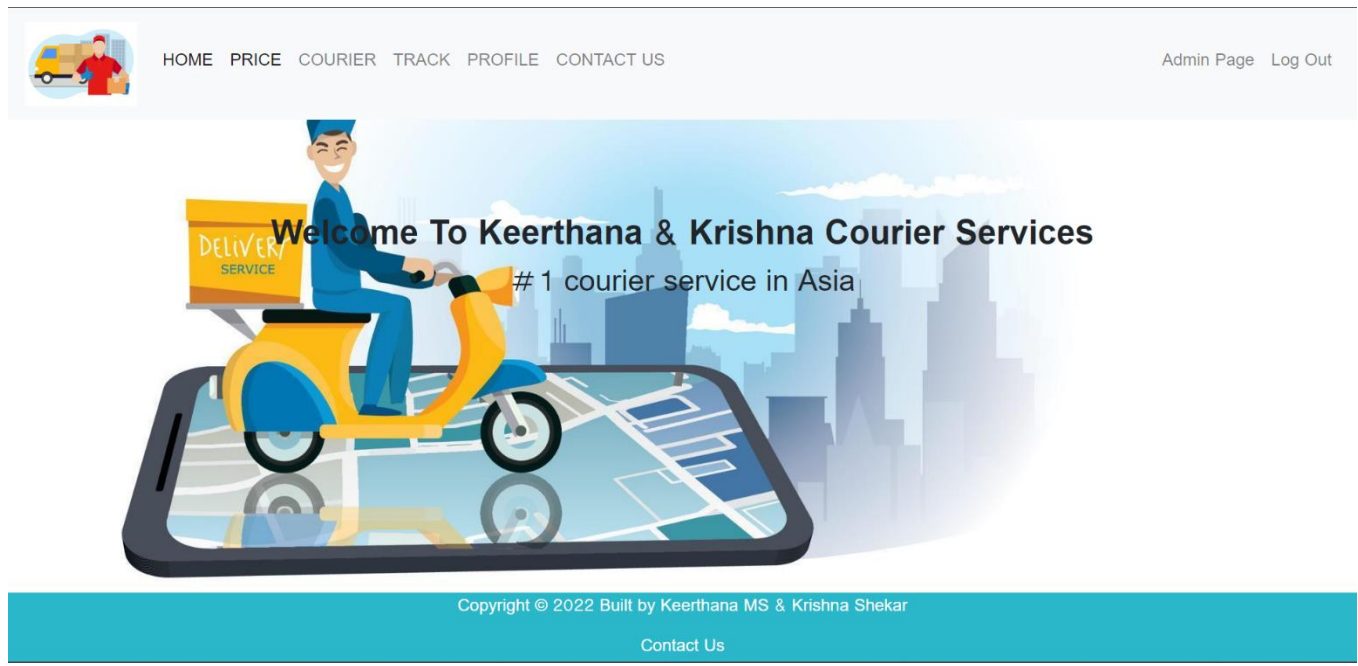


Fig 6. 2 Home Page

The fig 6.2 is the snapshot of the Home Page of the project, on the top of the page you can find the options for home, price, courier, track, profile, and contact us page. On the top left corner you can find login for admin page and log out option.

WEIGHT (in Kg)	PRICE
0-1	200
1-2	250
2-4	300
4-5	350
5-7	400
7-above	500

CHOOSE THE WEIGHT CATEGORY AND PAY AS PER:

PAY TO THE BELOW UPI ID

1. UPI: 123456789@okaxis
2. GPay: keerthana@gmail.com
3. PhonePay: 987654321

Fig 6.3 Pricing Of Courier

Fill The Details Of Sender & Receiver

SENDER		RECEIVER	
Name:	Keerthana MS	Name:	Krishna Shekar
Email:	keerthanams@gmail.com	Email:	krishna@gmail.com
PhoneNo.:	8789765456	PhoneNo.:	76894647383
Address:	Bengaluru	Address:	Mysuru
Weight:	78	Payment Id:	5643
Date:	01/09/2023	Items Image:	Choose File No file chosen

Place Order

Copyright © 2022 Built by Keerthana MS & Krishna Shekar

Contact Us

Fig 6.4 Courier Sending Page

NO.	ITEM IMAGE	SENDER NAME	RECEIVER NAME	RECEIVER EMAIL	ACTION
1		Keerthana MS	Krishna Shekar	krishna@gmail.com	Edit Delete CheckStatus
2		Keerthana MS	Deepthi	deepthi@gmail.com	Edit Delete CheckStatus

Copyright © 2022 Built by Keerthana MS & Krishna Shekar

Contact Us

Fig 6.5 Track Consignment Page



Fig 6.6 Tracking Status Of Parcel

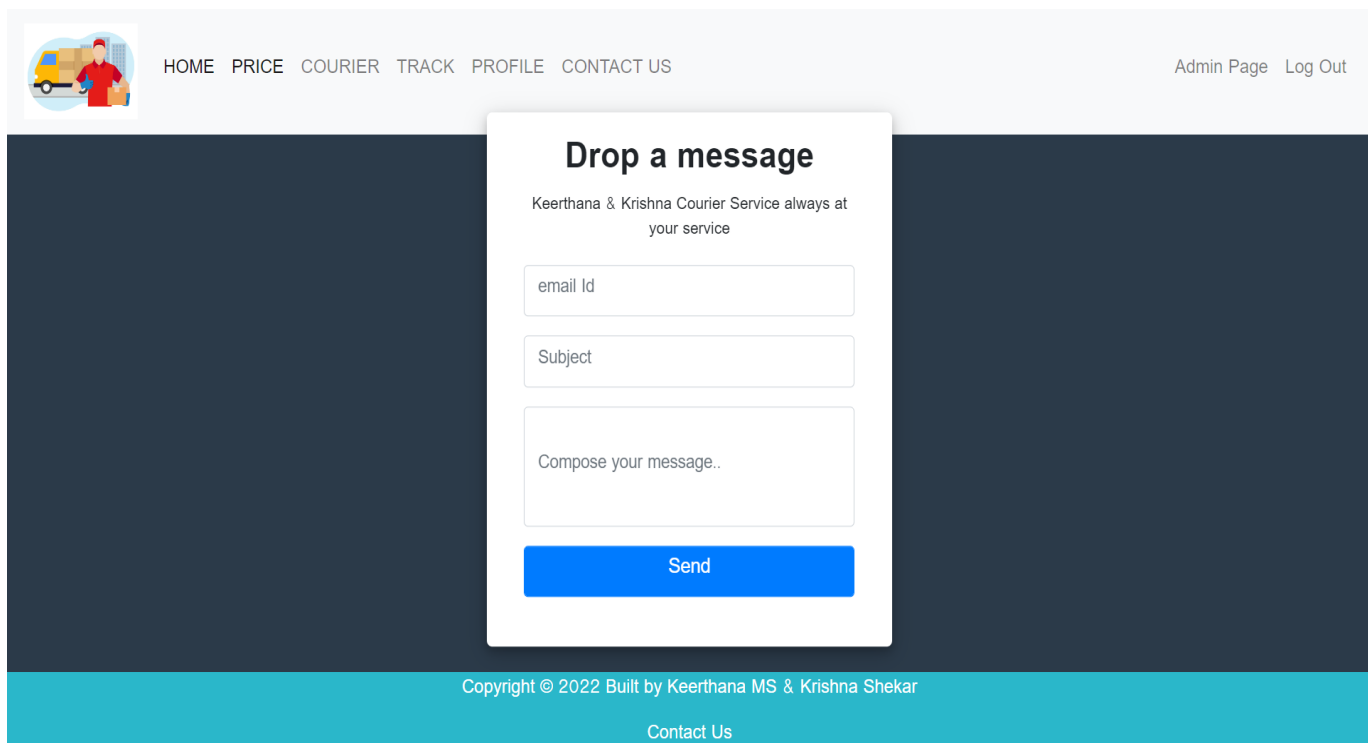


Fig 6.7 Contact Us Section

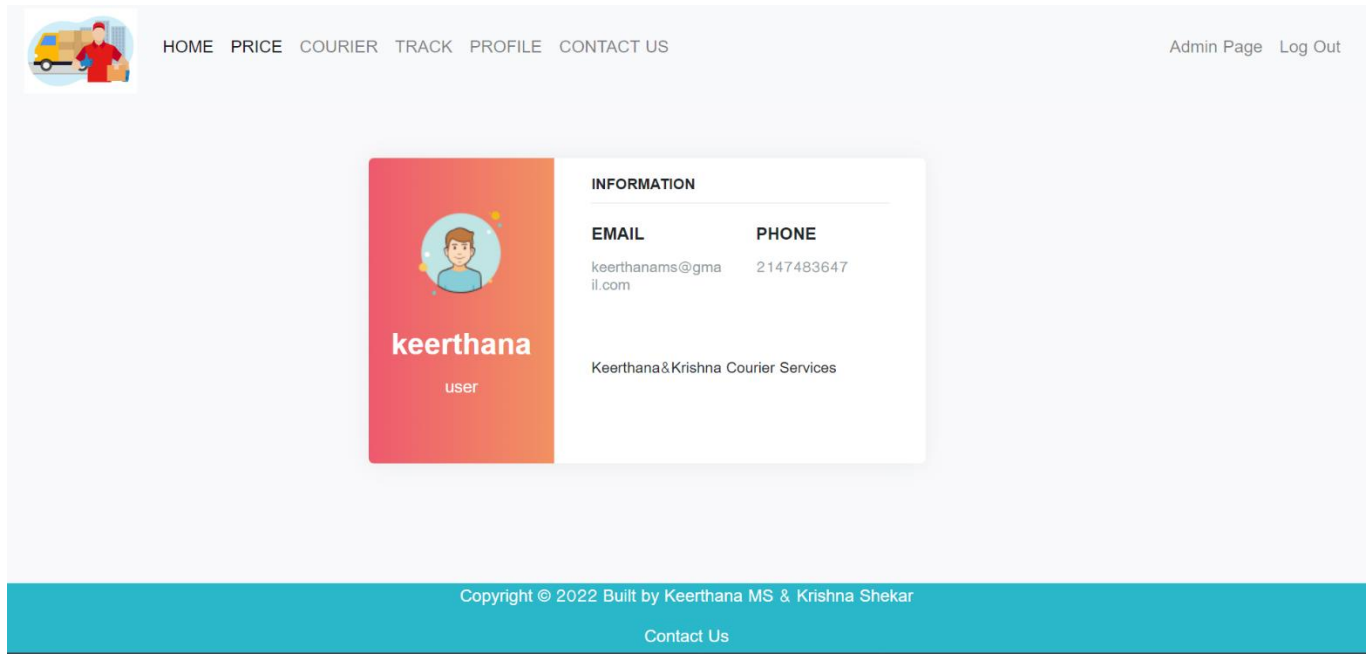
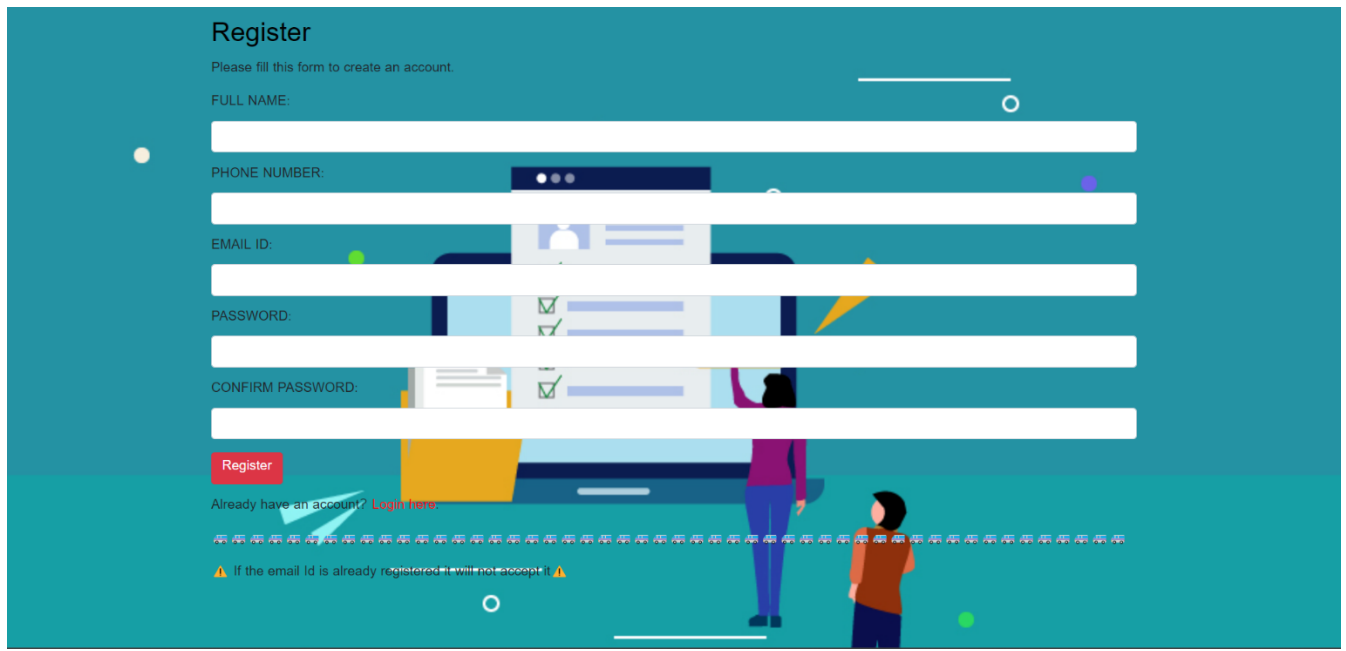


Fig 6.8 Profile View Section



The image shows a 'Register' form on a teal background with a faint illustration of people and a laptop. The form includes fields for 'FULL NAME', 'PHONE NUMBER', 'EMAIL ID', 'PASSWORD', and 'CONFIRM PASSWORD'. A red 'Register' button is at the bottom left. Below the button, there is a link 'Already have an account? Login here.' and a warning message: '⚠ If the email Id is already registered it will not accept ⚠'.

Register

Please fill this form to create an account.

FULL NAME:

PHONE NUMBER:

EMAIL ID:

PASSWORD:

CONFIRM PASSWORD:

Register

Already have an account? [Login here.](#)

⚠ If the email Id is already registered it will not accept ⚠

Fig 6.10 Register New Users Page

[Back To Home](#)

Admin Login

welcome Admin

Email_ID:

Password:

Fig 6.11 Admin Login Page

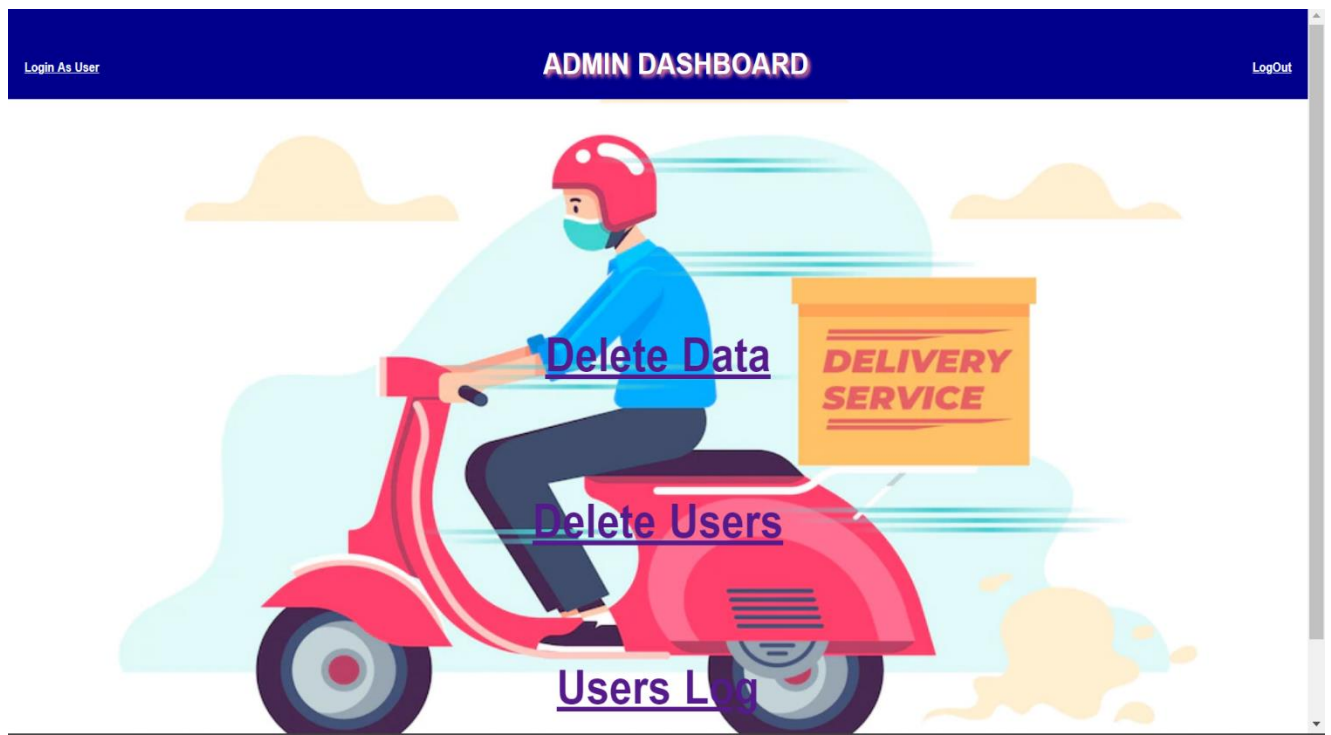


Fig 6.12 Admin Page







Search Data Information					
Back To Dashboard					
Log Out					
NO.	ITEM IMAGE	SENDER NAME	RECEIVER NAME	SENDER EMAIL	ACTION
1		krishna	susan	susan@gmail.com	Delete
2		keerthana	susan	keerthana@gmail.com	Delete
3		krishna	susan	krishna@gmail.com	Delete
4		Deeksha	Dayanand	deeksha@gmail.com	Delete
5		Keerthana MS	Krishna Shekar	keerthanams@gmail.com	Delete
6		Keerthana MS	Deepthi	keerthanams@gmail.com	Delete

Fig 6.13 Admin's Delete Data Page

[Back To Dashboard](#)

ALL USERS IN DATABASE

[Log Out](#)

NO.	USER NAME	EMAIL ID	ACTION
1	susan	susan@gmail.com	Delete User
2	keerthana	keerthana@gmail.com	Delete User
3	lizzie	lizzie@gmail.com	Delete User
4	krishna	krishna@gmail.com	Delete User
5	Deeksha	deeksha@gmail.com	Delete User
6	sudharshan	sudharshan@gmail.com	Delete User
7	keerthana	keerthanams@gmail.com	Delete User

Fig 6.14 All Users In Database

[DASHBOARD](#)

LOGS TABLE

[Log Out](#)

NO.	USER ID	ACTION TIME	ACTION PERFORMED	ACTION PERFORMED BY
1	104	2022-12-19 01:09:29	USER DELETED	USER
2	1	2022-12-19 01:09:43	USER DELETED	USER
3	105	2022-12-21 18:02:07	USER CREATED	USER
4	102	2022-12-21 18:11:57	USER DELETED	USER
5	14	2022-12-21 18:13:06	USER CREATED	USER
6	106	2023-01-09 19:00:37	USER CREATED	USER
7	107	2023-01-09 19:20:21	USER CREATED	USER
8	109	2023-01-09 19:28:54	USER CREATED	USER

Fig 6.15 User Log Table

UPDATE THE DETAILS

SENDER

RECEIVER

Name:

Keerthana MS

Name:

Krishna Shekar

Email:

keerthanams@gmail

Email:

krishna@gmail.com

PhoneNo.:

6574839376

PhoneNo.:

2345675849

Address:

Bengaluru

Address:

Mysuru

Weight:

12

ReceiptNo.:

6789

Date:

01/09/2023

Items Image:

Choose File No file chosen

Update

Fig 6.16 Update Courier Details Page

The screenshot shows the phpMyAdmin web interface. The top navigation bar includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, and More. The left sidebar shows a tree view of the database structure, with 'courierdb_sql' selected. The main content area displays the 'Structure' view for the 'courierdb_sql' database. It shows a table with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The tables listed are adlogin, admin, contacts, courier, login, logss, and users. At the bottom, there is a summary row for 7 tables and a console area.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> adlogin	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> admin	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> contacts	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> courier	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> login	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> logss	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	32.0 KiB	-
7 tables	Sum	33	InnoDB	utf8mb4_general_ci	208.0 KiB	0 B

At the bottom of the interface, there is a console area with a 'Console' tab and a 'Data dictionary' link.

Fig 6.17 List Of All The Tables In Project

CHAPTER 7

CONCLUSION

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

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

Following modifications or upgrades can be done in the system.

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

BIBLIOGRAPHY

- Text Book: 1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe.
7th Edition, 2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill
- YouTube: 1. https://www.youtube.com/watch?v=OK_JCrrv-c [PHP Programming]
2. <https://www.youtube.com/watch?v=Q8NPQ2RgWyg> [Front End Development]
3. <https://www.youtube.com/watch?v=LGTbdjoEBVM> [SQL]