**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction To The Project**

This Courier Management System Project will have different modules. The login section will

have login 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 is placing the orders

and all the details for the recipient such as its address, name, mobile number. During 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 internet. It

will provide 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 can not 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 profit

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 More Speed 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 Less Than Dbms In Certain Situations.
* Design Is Simple.
* Customization Is Easy And Efficient.

**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 lead 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 authorised users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorised 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 : Chorme, Firefox, Safari etcOperating 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**

Mainly 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. Eg. 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 Superkey), 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
   * 1. Email
     2. Password
     3. A\_Id
2. ADMIN
   * 1. A\_Id
     2. Email
     3. Name
     4. Pnumber
3. CONTACTS
   * 1. Id
     2. Email
     3. Subject
     4. Msg
4. COURIER
   * 1. C\_Id
     2. U\_Id
     3. Semail
     4. Remail
     5. Sname
     6. Rname
     7. Sphone
     8. Rphone
     9. Saddress
     10. Raddress
     11. Weight
     12. Billno
     13. Image
     14. Date
5. LOGIN
   * 1. Email
     2. Password
     3. U\_Id
6. LOGSS
   * 1. User\_Id
     2. Action\_Time
     3. Action\_Performed
     4. Action\_Performed\_By
7. USERS
   * 1. U\_Id
     2. Email
     3. Name
     4. Pnumber

**3.3 ENTITY RELATIONSHIP DIAGRAM**

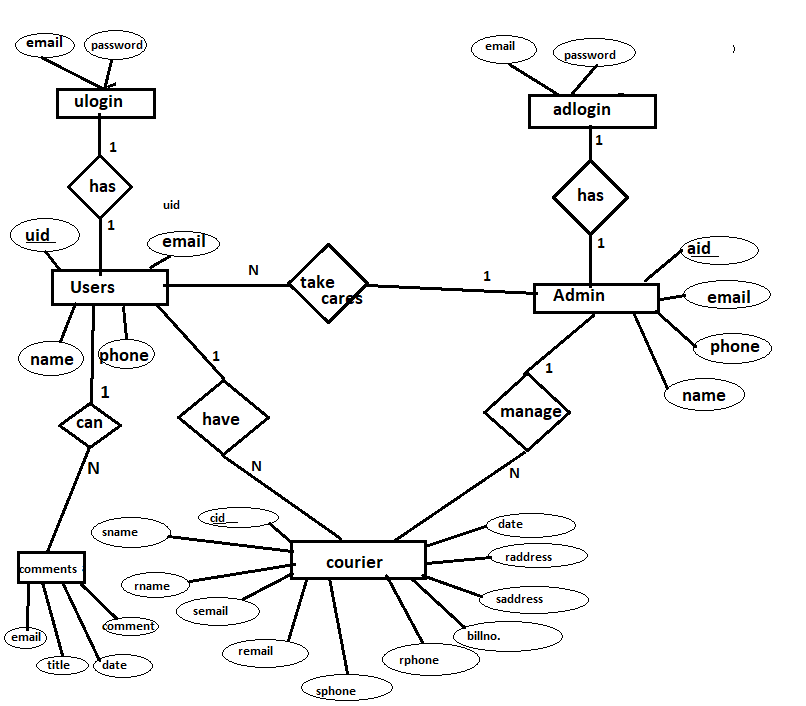
An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model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

**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**

Entity is the central concept of the Entity-Relationship model. An entity represents a description of the common features of set of objects of 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.

Diagram

Description automatically generated

**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! 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 👋");

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">Need For Speed 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>

</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 Need For Speed Courier Management Service</b></h2>

<h3>The fastest courier service of India</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');

if($dbcon==false)

{

echo "Database is not Connected!";

}

?>

**4.3 BACKEND CODE**

-- phpMyAdmin SQL Dump

-- version 5.1.1

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Jan 21, 2022 at 01:58 PM

-- Server version: 10.4.22-MariaDB

-- PHP Version: 7.4.27

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

-- Database: `courierdb`

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

-- 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

('admin1@gmail.com', '12345', 1),

('admin2@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, 'admin1@gmail.com', 'Admin1', 12345),

(2, 'admin2@gmail.com', 'Admin2', 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, 'mavs\_1@gmail.com', 'LOST', 'MY package is lost'),

(103, 'ishu@gmail.com', 'Package Damage', 'MY package is damaged, do contact me ASAP');

-- --------------------------------------------------------  
-- 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,

`remail` 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, 'anas@gmail.com', 'ishan@gmail.com', 'anas', 'ishaan', '123456', '654321', 'goa', 'noida', 5, 12, 'cddd.jpeg', '2022-01-18'),

(16, 13, 'mavs\_1@gmail.com', 'shanu@gmail.com', 'maviya', 'shanu', '987654321', '987123456', 'delhi', 'pune', 5, 11, '2959860-science-fiction-artwork-robot\_\_\_abstract-wallpapers.jpg', '2022-01-21'),

(17, 105, 'ishu@gmail.com', 'lisa@gmail.com', 'ishaan', 'lisa', '2147483647', '2474851756', 'noida', 'indore', 4, 10, 'fc.png', '2022-01-21');

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

-- 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

('anas@gmail.com', 'anas1', 12),

('mavs\_1@gmail.com', 'mav1', 13),

('ishu@gmail.com', 'ishaan1', 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, '2022-01-19 01:09:29', 'USER DELETED', 'USER'),

(1, '2022-01-19 01:09:43', 'USER DELETED', 'USER'),

(105, '2022-01-21 18:02:07', 'USER CREATED', 'USER'),

(102, '2022-01-21 18:11:57', 'USER DELETED', 'USER'),

(14, '2022-01-21 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, 'anas@gmail.com', 'mohd anas', 123456789),

(13, 'mavs\_1@gmail.com', 'maviya hassan', 987654321),

(14, 'javed@gmail.com', 'S javed', 2118529630),

(105, 'ishu@gmail.com', 'ishaan', 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.](https://en.wikipedia.org/wiki/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,](https://en.wikipedia.org/wiki/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 behaviour 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**

Text

Description automatically generated with medium confidence

**Fig 6.1 User Login Page**

Graphical user interface

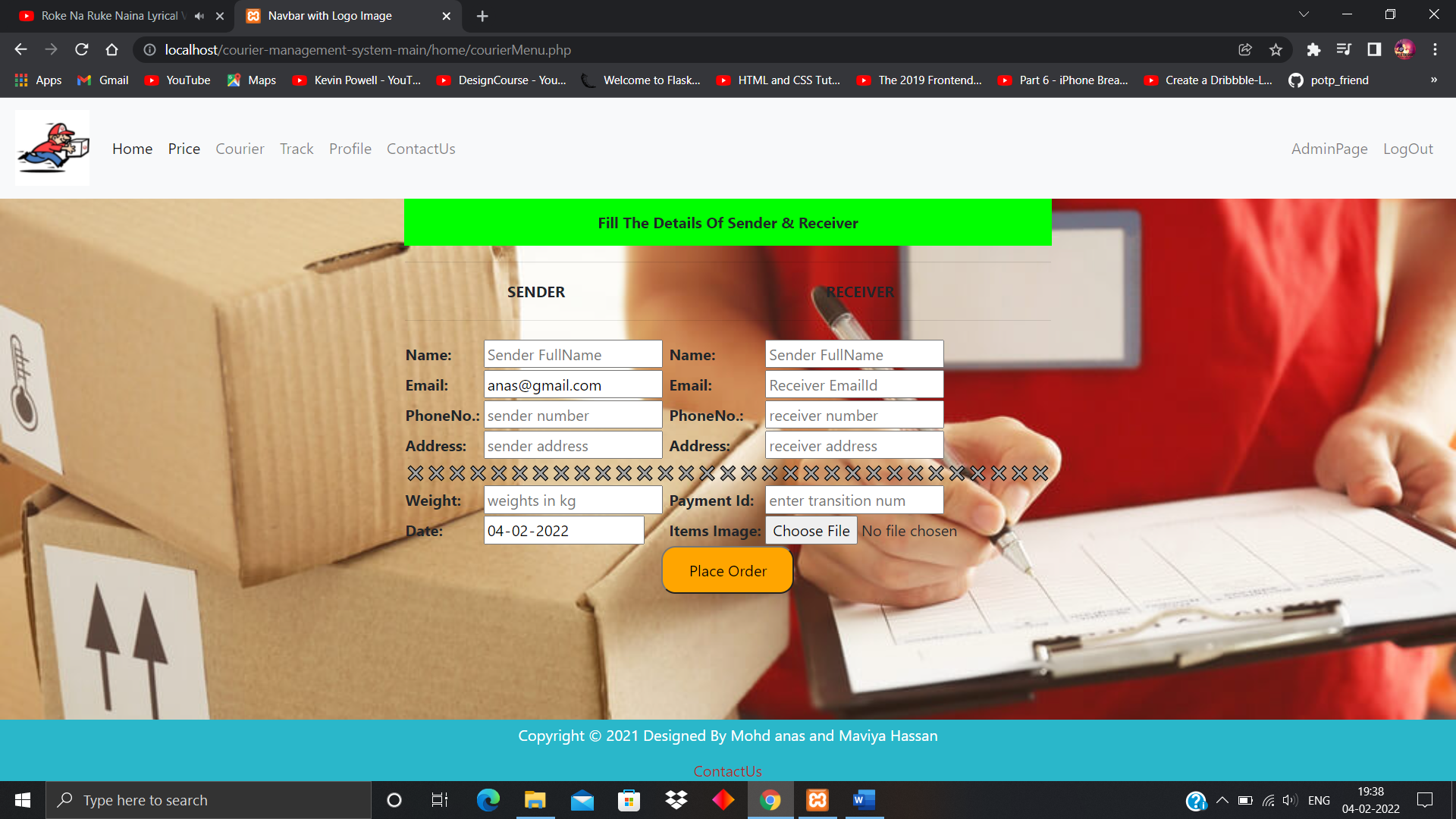
Description automatically generated

**Fig 6. 2 Home Page**

Graphical user interface, text

Description automatically generated

**Fig 6.3 Pricing Of Courier**



**Fig 6.4 Courier Sending Page**

A screenshot of a computer

Description automatically generated

**Fig 6.5 Track Consignment Page**

Graphical user interface, text, website

Description automatically generated

**Fig 6.6 Track Status Of Parcel**

A screenshot of a computer

Description automatically generated

**Fig 6.7 Contact Us Section**

Graphical user interface, application

Description automatically generated

**Fig 6.8 Profile View Section**

Graphical user interface, application

Description automatically generated

**Fig 6.9 Profile View Section**

A screenshot of a computer

Description automatically generated

**Fig 6.10 Register New Users Page**

Graphical user interface, website

Description automatically generated

**Fig 6.11 Admin Login Page**

Graphical user interface, application, website

Description automatically generated **Fig 6.12 Admin Page**

Graphical user interface

Description automatically generated with medium confidence

**Fig 6.13 Admin’s Delete Data Page**

Table

Description automatically generated

**Fig 6.14 All The User Details Page**

A screenshot of a computer

Description automatically generated

**Fig 6.15 All User’s Related Log’s Page**

A screenshot of a computer

Description automatically generated

**Fig 6.16 Update Courier Details Page**

A screenshot of a computer

Description automatically generated

**Fig 6.17 List Of All The Tables In Project**

**CHAPTER 7**

**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

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. Text Book: Ramez Elmasri and Shamkant B. Navathe “Fundamentals of Database Systems” ,7th Edition , Pearson , 2017.
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