

A
MINI PROJECT REPORT
ON

**COURIER MANAGEMENT
SYSTEMS**

Of
Database Management System Lab

SUBMITTED BY:

1. Uday Singh Slathia (RA2111003011085)
2. Hrithik (RA2111003011086)

UNDER THE GUIDANCE OF
Dr.B.Muruganantham
(Associate Professor, Department of Computing Technologies)

**Department Of Computing Technologies,
School of Computing
SRM Institute of Science and Technology.**

Academic Year 2023- 24





SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this project report "**Courier Management System**" is the bonafide work of "**Uday Singh Slathia (RA2111003011085),Hrithik (RA2111003011086)**" of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC303J- Database Management systems in SRM Institute of Science and Technology during the academic year 2023-2024(Even sem).

SIGNATURE

Dr. B.Muruganantham

Associate Professor

Computing Technologies

SIGNATURE

Dr. M. Pushpalatha

HOD CTECH

Computing Technologies

ABSTRACT

Courier Service Delivery Management System is helpful for little and enormous scale messenger services for up their services and increasing sales and services by victimization net based mostly portal. Users will book messenger, fathom arrival of packages, closing and gap timings, client support and feedback system for finishing add less time. At this time most of the big scale messenger services in developed countries use this technology. Existing messenger Service corporations in developing nations like Nigeria work on the manual management methodology that may be a time overwhelming method. Hence, the aim of this work is to style and implement a National messenger Management System capable of enhancing client deliveries by options like speed, security, following etc. from specific cities or cities, to regional and national services. The messenger Service Delivery Management System are divided into 2 users that as the ‘Administrators’ and ‘Customers’. This work are designed and enforced employing a 3 tier application style approach. The look of the graphical program are designed with machine-readable text Mark-UP Language (HTML); MySQL is that the information of selection whereas PHP language are wont to connect the user to the knowledge on the information. The Methodology are supported SDLC (Systems Development Life Cycle) body of water model. The phases as designing, Analysis, style and Impementation. The expected output is to be interactive web-based, that may solve the issues of the manual system of record management.

Table of Contents

S.No	Chapter	Page No
1	Introduction	4
2	Scope	5
3	Requirement Analysis 4.1 Functional Requirement 4.2 Non Functional Requirement	6
4	Data Modeling a. ER Diagram b. Relational model c. Normalization	7-9
5	Software Requirements a. Front End b. Back End c. Database Connectivity	10-11
6	Sample Source Code	12-20
7	Sample Screen shots	21-26
8	Conclusion	27
9	References	28

Introduction

A courier is a person who makes arrangements for or accompanies a group of travelers on a journey or tour(William,2009). In ancient history runners and orienting pigeon sand riders on horseback were accustomed deliver timely messages. Before there have been mechanized traveler services foot messengers physically ran miles to their destinations. To this day there are marathons directly related to actual historical traveler routes. Within the Middle Ages, royal courts maintained their own messengers United Nations agency were paid very little over common laborers. In cities, there as usually bicycle couriers or bike couriers except for consignments requiring delivery over larger distance networks, this may often embrace lorries, railways and craft. Many corporations United Nations agency operate below a Just-In-Time or "JIT" inventory technique typically utilize on-board couriers. On-board couriers square measure people United Nations agency will travel at a moment's notice anyplace within the world, typically via airlines. Whereas this kind of service is that the second costliest—general aviation charters square measure much more expensive—companies associate degree Alyce the value of service to interact an on-board messenger versus the "cost" the corporate can notice ought to the merchandise not arrive by a such that time (i.e. AN mechanical system stopping, untimely court filing, lost sales from product or elements missing a delivery point, organ transplants) (Small, 1990). Package delivery or parcel delivery is the delivery of shipping containers, parcels, or high value mail as single shipments. The service is provided by most communication systems, mail, non-public package delivery services, and fewer than truckload shipping carriers. Continued growth of business-to-consumer (b2c) e-commerce has increased demand for low-cost package shipping services. Demand for cheap parcel shipping is very intense for on-line and catalog retailers. The "cost" the corporate can understand ought to the merchandise not arrive by a nominative time These merchants, several of whom primarily ship low-priced product, face shoppers proof against paying immoderate shipping prices (often driven up by fuel surcharges, residential delivery fees, etc.) for package delivery to their homes. As a result, package shipping consolidators step in to mix inexpensive "last-mile delivery" strengths of the United States of America mail service with the technological and operational capabilities usually related to nonpublic carriers.

Scope

The scope of a courier management system (CMS) typically encompasses various aspects of managing the process of delivering packages and parcels from one location to another efficiently and effectively. Here's a breakdown of the key components and functionalities that are usually included in such a system:

User Management: This involves managing different types of users such as customers, couriers, and administrative staff. Each user may have different levels of access and permissions within the system.

Order Management: Users should be able to place orders for package delivery through the system. This includes specifying pickup and delivery addresses, package dimensions, weight, delivery preferences (e.g., express delivery), and tracking information.

Dispatch Management: Dispatchers or automated algorithms manage the assignment of delivery tasks to couriers based on factors like proximity, availability, and workload. This includes optimizing routes to ensure timely deliveries and minimize fuel consumption.

Tracking and Visibility: Customers and administrators should be able to track the status and location of packages in real-time. This includes providing tracking numbers, delivery updates, and estimated delivery times.

Reporting and Analytics: The system should provide comprehensive reporting and analytics capabilities to track key performance indicators (KPIs) such as delivery times, delivery accuracy, customer satisfaction, and courier performance.

Inventory Management: If applicable, the CMS may include features for managing inventory in warehouses or distribution centers, including stock tracking, replenishment, and order fulfillment.

Customer Service and Support: The system should provide channels for customer support, such as live chat, email, or phone support. It should also have features for handling customer inquiries, complaints, and returns.

Integration and Scalability: The CMS should be designed to integrate with other systems and services such as e-commerce platforms, ERP systems, and transportation management systems (TMS). It should also be scalable to accommodate growth in the volume of packages and users.

Security and Compliance: The system should adhere to security best practices to protect sensitive customer data and payment information. It should also comply with relevant regulations such as GDPR (General Data Protection Regulation) and PCI DSS (Payment Card Industry Data Security Standard).

The specific features and functionalities of a courier management system may vary depending on the needs of the organization implementing it and the scale of its operations. Additionally, advancements in technology such as artificial intelligence (AI) and Internet of Things (IoT) may enable further enhancements and automation in courier management processes.

Requirement Analysis

Functional Requirements:

Functional requirements describe the specific behaviors and functionalities that the CMS should exhibit. They define what the system should do. Here are some examples of functional requirements for a courier management system:

User Registration and Authentication: Users should be able to register for accounts and log in securely to access the system.

Order Placement: Customers should be able to place orders for package delivery, specifying pickup and delivery addresses, package details, and delivery preferences.

Dispatching and Routing: The system should automatically assign delivery tasks to couriers based on factors such as proximity, availability, and workload. It should also optimize delivery routes to minimize time and cost.

Package Tracking: Customers and administrators should be able to track the status and location of packages in real-time using tracking numbers or other identifiers.

Reporting and Analytics: The system should provide reporting and analytics features to track key performance indicators (KPIs) such as delivery times, customer satisfaction, and courier performance.

Inventory Management: If applicable, the system should include features for managing inventory in warehouses or distribution centers, including stock tracking and order fulfillment.

Non-Functional Requirements:

Non-functional requirements describe the qualities or attributes that the CMS should possess. They define how the system should perform. Here are some examples of non-functional requirements for a courier management system:

Performance: The system should be able to handle a large volume of transactions and users simultaneously without significant performance degradation.

Reliability: The system should be reliable and available 24/7, with minimal downtime for maintenance or upgrades.

Security: The system should adhere to security best practices to protect sensitive data and prevent unauthorized access or breaches.

Usability: The system should be intuitive and easy to use, with a user-friendly interface that minimizes the learning curve for users.

Scalability: The system should be scalable to accommodate growth in the volume of packages and users without compromising performance or reliability.

Compatibility: The system should be compatible with various devices, operating systems, and web browsers to ensure a seamless user experience.

Regulatory Compliance: The system should comply with relevant regulations and standards such as GDPR, PCI DSS, and industry-specific regulations for the courier and logistics industry.

Data Modeling

ER Diagram

The below mentioned Entity-Relationship diagram (ER -diagram) describes the structure of the Courier management system database.

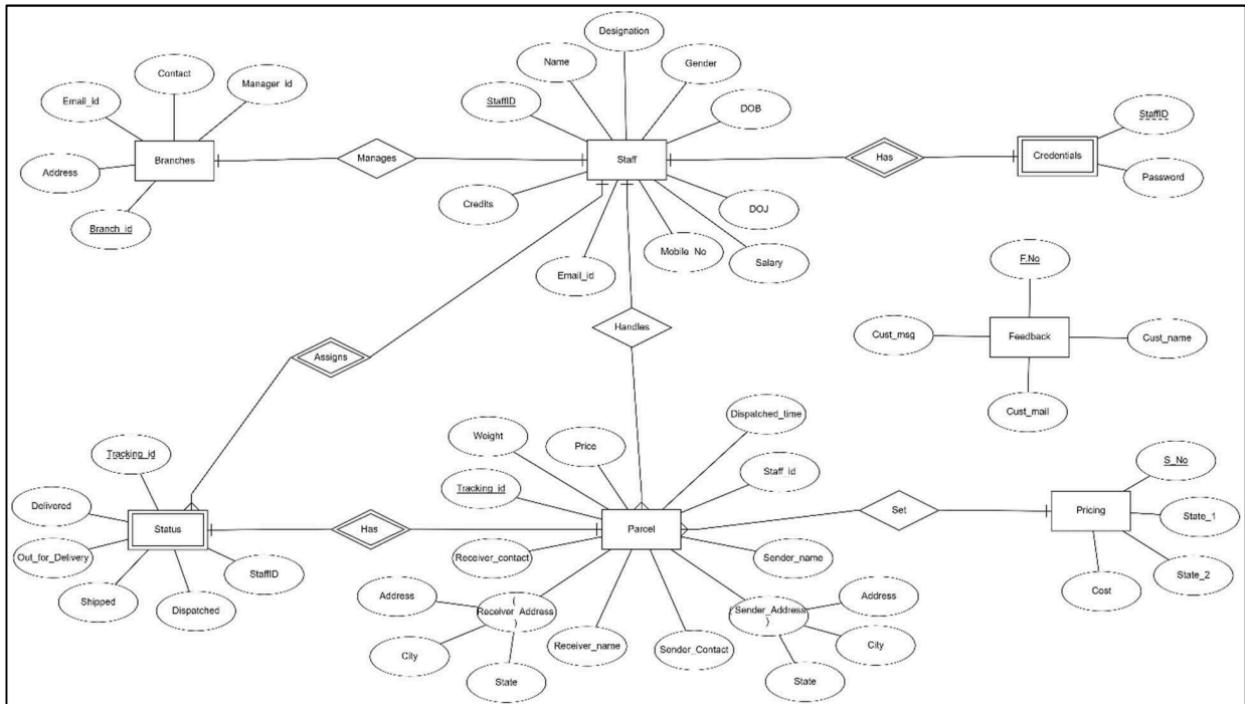


Fig 1 : Entity-relationship diagram of the database

DESCRIPTION OF THE ER MODEL:

The Courier Management System database has various entities namely Staff, Credentials, Branches, Parcel, Status, Pricing and Feedback.

Staff entity stores the details of the staff who work for the courier management service. It has a one-to-one relationship with the Credentials entity as every staff member has a unique staff_id and a password and every credential refers to a unique staff. The staff entity also holds a one-to-one relationship with the Branches entity because each branch is managed by a unique staff and each staff manages a unique branch. The staff entity holds a one-to-many relationship with the Parcel entity since a staff can handle many parcels while a parcel could only be handled by a single staff. The staff entity also holds a one-to-many relationship with the Status entity as a staff can assign the status of any number of parcels whereas a status of a parcel could only be assigned by a single staff member. The Parcel entity has a many-to-one relationship with the Pricing entity as pricing could be set to any number of parcels and they may have the same price whereas a single parcel could have only one pricing. The Parcel entity has a one-to-one relationship with the Status entity as a single parcel could have a single status and a single status of a tracking_id could have a unique parcel.

Relational model

The Relational model mentioned below is a skeleton structure that represents the logical view of the Courier Management System database as a whole. This database schema design organizes the data into separate entities, determines how to create relationships between organized entities, and how to apply the constraints on the data.

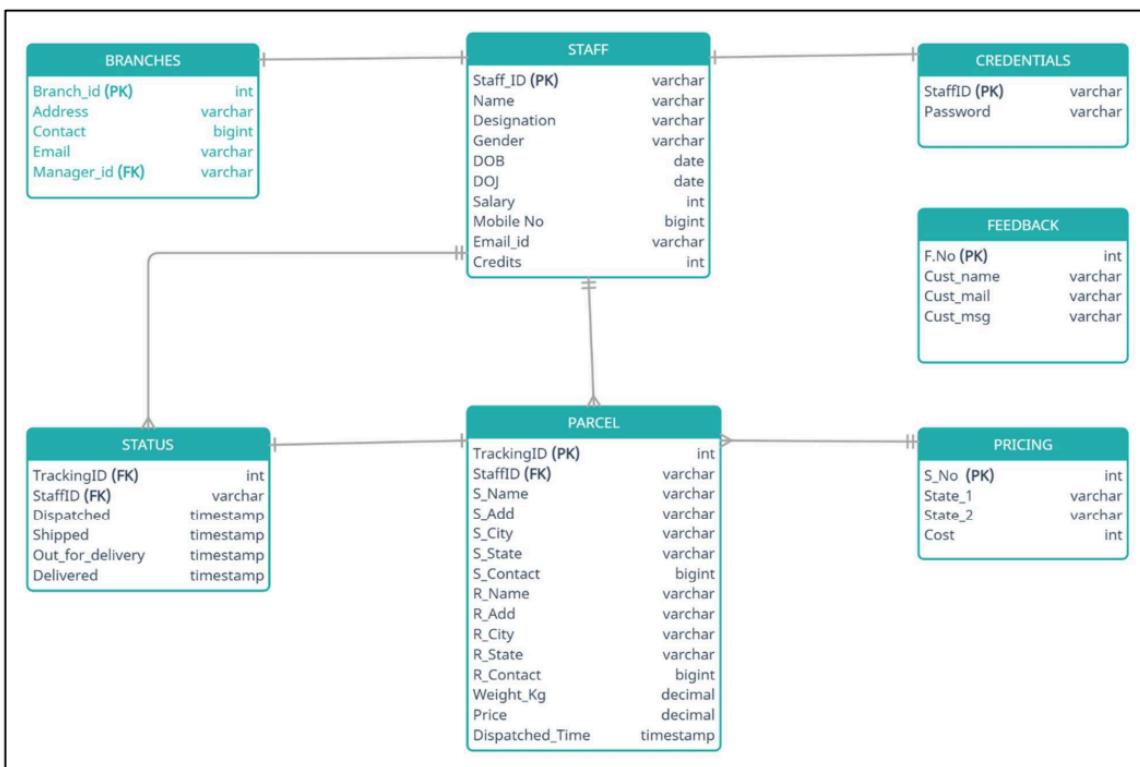


Fig 2 : Relation schema model

The Relational model elucidates about the relationship each table is having with others. It also says about how the primary key and foreign key constraints are used in order to provide a relationship between the tables. All the tables have been normalized and are present in Boyce – Codd Normal Form since all the transitive functional dependencies have been removed by decomposing the tables.

Normalization

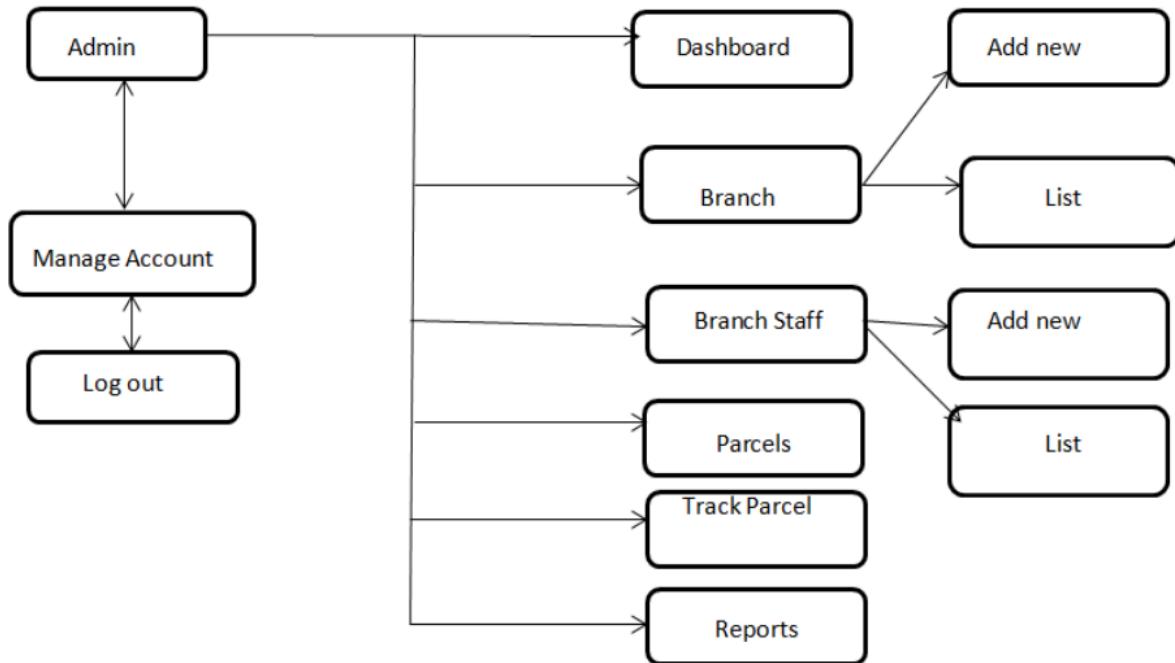


Fig 3: Normalization Of The System

The above diagram shows the flow, once the user logs in permitting that individual user to settle on the item from the given menu and send them to the ultimate order page and ensure the order, and additionally displays the given order

Software Requirements

Front End Requirements:

User Interface: The front end should have a user-friendly interface for customers, couriers, and administrators to interact with the system.

Responsive Design: The interface should be responsive and compatible with different devices, including desktops, tablets, and smartphones.

Navigation: Intuitive navigation menus and buttons should facilitate easy access to different functionalities and pages within the system.

Forms and Input Validation: Forms for order placement, user registration, and other interactions should include proper validation to ensure data integrity.

Feedback and Notifications: The front end should provide feedback messages and notifications to users about the status of their actions, such as successful order placement or errors during form submission.

Accessibility: The interface should adhere to accessibility standards to ensure that users with disabilities can access and use the system effectively.

Customization: Users may have the option to customize their preferences, such as language selection or theme colors.

Back End Requirements:

Server-Side Logic: The back end should handle business logic and process requests from the front end, including order placement, dispatching, and tracking.

Security: Implement measures for user authentication, authorization, and data encryption to protect sensitive information.

Performance Optimization: Optimize server-side code and database queries to ensure fast response times and efficient use of resources.

Error Handling: Proper error handling mechanisms should be in place to handle exceptions and provide meaningful error messages to the front end.

Integration: The back end should integrate with external services and APIs for functionalities such as payment processing, geolocation, and communication.

Scalability: Design the back end to be scalable, allowing it to handle increased loads as the system grows.

Logging and Monitoring: Implement logging and monitoring tools to track system activities, diagnose issues, and analyze performance metrics.

Database Connectivity Requirements:

Data Storage: The database should store information such as user accounts, order details, package tracking data, and inventory.

Reliability: Ensure data integrity and reliability through mechanisms such as transactions, constraints, and backups.

Performance: Optimize database performance through proper indexing, query optimization, and data partitioning.

Scalability: Design the database to scale horizontally or vertically to accommodate increased data volume and user traffic.

Security: Implement security measures such as access control, encryption, and data masking to protect sensitive information stored in the database.

Data Migration: Plan for data migration strategies during system upgrades or migrations to new environments.

Compatibility: Ensure compatibility between the database management system (DBMS) and other components of the system, such as the back end and front end.

Language :PHP

Frontend : HTML,CSS Framework : Codeigniter

Database : M Y SQL

- **PHP**

PHP is a General-purpose Programming Language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by the PHP Group. PHP originally stood for Personal Home Page, but it now Stands for The recursive Initialism PHP

- **HTML**

HTML, machine-readable text terminology, provides content structure and that means by process that content as, as an example, headings, paragraphs, or images.

- **CSS**

CSS, or Cascading Style Sheets, is a presentation language created to style content's appearance—for example, fonts or colors. HTML helps in building a website and CSS for its styling.

- **JAVASCRIPT**

JavaScript is a fun and flexible programming language. It's one of the core technologies of web development and can be used on both the front-end and the back-end.

- **MySQL**

As we all know that we have a tendency to do need a information server to store any content of the user record or account management and simple record following. What higher possibility than SQL itself. a typical language for storing, manipulating, and retrieving knowledge in databases

Sample Source Code

This interactive web application is completely responsive thereby suitable for all screen sizes and user-friendly. The user interface is developed used HTML5, CSS3 and Bootstrap 4 which structures the application, styles its contents and develops mobile-friendly responsiveness respectively (Frontend development). The serverside scripting is implemented in PHP7, which is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. Along with MySQL database management system, which is a widely-used relational DBMS comprises of the Backend development stack. The PHP development environment, XAMPP, is a free and open-source crossplatform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP v.3.2.3 used in this project provides Apache web server and MySQL database.

The various stages of the implementation of this project are briefed as follows,

1. CREATING RELATIONS :

The various relations(tables) used to represent the data in the database include :

```
STAFF (StaffID, Name, Designation, Gender, DOB, DOJ, Salary,  
Mobile, Email, Credits)  
CREDENTIALS (StaffID, Pwd)  
BRANCHES (Branch_id, Address, Contact, Email, Manager_id)  
PARCEL (TrackingID, StaffID, S_Name, S_Add, S_City, S_State,  
S_Contact, R_Name, R_Add, R_City, R_State, R_Contact,  
Weight_Kg, Price, Dispatched_Time)  
PRICING (S.No, State_1, State_2, Cost)  
STATUS (TrackingID, StaffID, Dispatched, Shipped,  
Out_for_delivery, Delivered)  
FEEDBACK ( F.No, Cust_name, Cust_mail, Cust_msg)
```

Server: 127.0.0.1 » Database: cc_couriers

Structure SQL Search Query Export Import Operations Privileges

Run SQL query/queries on database cc_couriers:

```

1 CREATE TABLE `credentials` (`StaffID` varchar(30) NOT NULL, `Pwd` varchar(30) NOT NULL);
2
3 CREATE TABLE `staff` (`StaffID` varchar(30) NOT NULL, `Name` varchar(30) NOT NULL,
4     `Designation` varchar(30) NOT NULL, `Gender` varchar(10) NOT NULL,
5     `DOB` date NOT NULL, `DOJ` date NOT NULL, `Salary` int(11) NOT NULL,
6     `Mobile` bigint(20) NOT NULL, `Email` varchar(30) NOT NULL,
7     `Credits` int(11) NOT NULL DEFAULT '0' );
8
9 CREATE TABLE `branches` (`Branch_id` int(11) NOT NULL, `Address` varchar(100) NOT NULL,
10    `Contact` bigint(20) NOT NULL, `Email` varchar(40) NOT NULL,
11    `Manager_id` varchar(30) NOT NULL);
12
13 CREATE TABLE `pricing` (`S.No` int(11) NOT NULL, `State_1` varchar(30) NOT NULL,
14     `State_2` varchar(30) NOT NULL, `Cost` int(11) NOT NULL);
15
16 CREATE TABLE `parcel` (`TrackingID` int(11) NOT NULL, `StaffID` varchar(30) NOT NULL,
17     `S_Name` varchar(30) NOT NULL, `S_Add` varchar(50) NOT NULL,
18     `S_City` varchar(20) NOT NULL, `S_State` varchar(20) NOT NULL,
19     `S_Contact` bigint(20) NOT NULL, `R_Name` varchar(30) NOT NULL,
20     `R_Add` varchar(50) NOT NULL, `R_City` varchar(20) NOT NULL,
21     `R_State` varchar(20) NOT NULL, `R_Contact` bigint(20) NOT NULL,
22     `Weight_Kg` decimal(10,2) NOT NULL, `Price` decimal(10,2) NOT NULL,
23     `Dispatched_Time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP);
24
25 CREATE TABLE `status` (`TrackingID` int(11) NOT NULL, `StaffID` varchar(30) NOT NULL,
26     `Dispatched` timestamp NULL DEFAULT NULL,
27     `Shipped` timestamp NULL DEFAULT NULL,
28     `Out_for_delivery` timestamp NULL DEFAULT NULL,
29     `Delivered` timestamp NULL DEFAULT NULL);
30
31 CREATE TABLE `feedback` (`F.No` int(11) NOT NULL, `Cust_name` varchar(30) NOT NULL,
32     `Cust_mail` varchar(50) NOT NULL, `Cust_msg` varchar(500) NOT NULL);
33

```

Fig 4.1 : SQL queries to create tables

phpMyAdmin

Server: 127.0.0.1 » Database: cc_couriers

Structure SQL Search Query Export Import Operations Privileges Routines Events

New

- Events
- Tables
 - branches
 - credentials
 - feedback
 - parcel
 - pricing
 - staff
 - status
- Views
- frozen_bottle
- information_schema
- misc

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
branches	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	32 Kib	-
credentials	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 Kib	-
feedback	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 Kib	-
parcel	Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	32 Kib	-
pricing	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 Kib	-
staff	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	32 Kib	-
status	Browse Structure Search Insert Empty Drop	18	InnoDB	latin1_swedish_ci	16 Kib	-
7 tables	Sum	80	InnoDB	latin1_swedish_ci	160 Kib	0 B

Check all

Fig 4.2 : Created tables in the database

2. CREATING VIEWS :

A view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. Here two different views are created on the relations parcel and status.

The screenshot shows the MySQL Workbench interface with the following details:

- Server: 127.0.0.1
- Database: cc_couriers
- Toolbar buttons: Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines.
- Filters section: Containing the word: [empty input field].
- Table view:

Table	Action	Rows	Type	Collation	Size	Overhead
arrived	Browse Structure Search Insert Drop	~0	View	---	-	-
delivered	Browse Structure Search Insert Drop	~0	View	---	-	-
2 tables Sum				InnoDB latin1_swedish_ci	0 B	0 B
- Buttons at the bottom: Up, Check all, With selected: dropdown.

Fig 5.1 : List of views in the database

i. **Arrived view:** stores the details of consignments in transit.

The screenshot shows the MySQL Workbench interface with the following details:

- Server: 127.0.0.1
- Database: cc_couriers
- Toolbar buttons: Structure, SQL, Search, Query, Export, Import.
- SQL Editor title: Run SQL query/queries on database cc_couriers:
- SQL code:

```
1 CREATE OR REPLACE VIEW arrived AS
2 SELECT P.*, S.Shipped, S.Out_for_delivery, S.Delivered
3 FROM parcel P, status S
4 WHERE P.TrackingID = S.TrackingID AND S.Delivered is NULL;
```

Fig 5.2 : SQL query to create the view - Arrived

The screenshot shows the phpMyAdmin interface for the 'cc_couriers' database. On the left, there's a tree view of tables and views. Under 'Tables', 'arrived' is selected. The main area shows the 'Relation view' of the 'arrived' view. The view has 18 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	TrackingID	int(11)			No	0		
2	StaffID	varchar(30)	latin1_swedish_ci		No	None		
3	S_Name	varchar(30)	latin1_swedish_ci		No	None		
4	S_Add	varchar(50)	latin1_swedish_ci		No	None		
5	S_City	varchar(20)	latin1_swedish_ci		No	None		
6	S_State	varchar(20)	latin1_swedish_ci		No	None		
7	S_Contact	bigint(20)			No	None		
8	R_Name	varchar(30)	latin1_swedish_ci		No	None		
9	R_Add	varchar(50)	latin1_swedish_ci		No	None		
10	R_City	varchar(20)	latin1_swedish_ci		No	None		
11	R_State	varchar(20)	latin1_swedish_ci		No	None		
12	R_Contact	bigint(20)			No	None		
13	Weight_Kg	decimal(10,2)			No	None		
14	Price	decimal(10,2)			No	None		
15	Dispatched_Time	timestamp			No	0000-00-00 00:00:00		
16	Shipped	timestamp			Yes	NULL		
17	Out_for_delivery	timestamp			Yes	NULL		
18	Delivered	timestamp			Yes	NULL		

Fig 5.3 : Structure of the view - Arrived

ii. **Delivered view** : stores the details of the consignments delivered.

The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. A SQL query is entered in the 'Run SQL query/queries on database cc_couriers:' field:

```

1 CREATE OR REPLACE VIEW delivered AS
2 SELECT P.*, S.Shipped, S.Out_for_delivery, S.Delivered
3 FROM parcel P, status S
4 WHERE P.TrackingID = S.TrackingID AND S.Delivered is NOT NULL;

```

Fig 5.4 : SQL query to create view – Delivered

Server: 127.0.0.1 » Database: cc_couriers » View: delivered

Browse Structure SQL Search Insert Export Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	TrackingID	int(11)			No	0		
2	StaffID	varchar(30)	latin1_swedish_ci		No	None		
3	S_Name	varchar(30)	latin1_swedish_ci		No	None		
4	S_Add	varchar(50)	latin1_swedish_ci		No	None		
5	S_City	varchar(20)	latin1_swedish_ci		No	None		
6	S_State	varchar(20)	latin1_swedish_ci		No	None		
7	S_Contact	bigint(20)			No	None		
8	R_Name	varchar(30)	latin1_swedish_ci		No	None		
9	R_Add	varchar(50)	latin1_swedish_ci		No	None		
10	R_City	varchar(20)	latin1_swedish_ci		No	None		
11	R_State	varchar(20)	latin1_swedish_ci		No	None		
12	R_Contact	bigint(20)			No	None		
13	Weight_Kg	decimal(10,2)			No	None		
14	Price	decimal(10,2)			No	None		
15	Dispatched_Time	timestamp			No	0000-00-00 00:00:00		
16	Shipped	timestamp			Yes	NULL		
17	Out_for_delivery	timestamp			Yes	NULL		
18	Delivered	timestamp			Yes	NULL		

Check all With selected: Browse

Fig 5.5 : Structure of the view – Delivered

3. CREATING TRIGGERS :

A trigger is a stored procedure in the database which automatically invokes whenever a special event in the database occurs. placeParcel : Trigger to insert the values of the attributes -TrackingId and dispatched_time - of parcel relation into the status relation after inserting a new tuple in the parcel relation.

Server: 127.0.0.1 » Database: cc_couriers

Structure SQL Search Query Export Import

Run SQL query/queries on database cc_couriers:

```

1 DELIMITER $$ 
2 CREATE TRIGGER `placeParcel` AFTER INSERT ON `parcel` FOR EACH ROW BEGIN
3     UPDATE staff SET Credits=Credits+5 WHERE StaffID=NEW.StaffID;
4 
5     INSERT INTO status (TrackingID, StaffID, Dispatched)
6     VALUES ( NEW.TrackingID, NEW.StaffID, NEW.Dispatched_Time);
7 END
8 $$ 
9 DELIMITER ;

```

Fig 6.1 : SQL query to create the trigger - placeParcel

Name	Table	Action	Time	Event
<input type="checkbox"/> placeParcel	parcel	Edit Export Drop	AFTER	INSERT

Check all With selected: Export Drop

Fig 6.2 : Result of successful creation of the trigger

4. CREATING EVENTS :

MySQL Events are named objects which contain one or more SQL statement. They are stored in the database and executed at one or more intervals, say once every week or month. resetCredits: Event which resets the credits received by each staff to 0 at the beginning of every month in order to award - the Employee of the Month.

```

1 DELIMITER $$ 
2 CREATE EVENT `resetCredits` ON SCHEDULE EVERY 1 MONTH STARTS '2021-05-01 00:00:00'
3 ON COMPLETION NOT PRESERVE ENABLE DO BEGIN
4     update staff set Credits=0;
5 END
6 $$ 
7 DELIMITER ;

```

Fig 7.1 : SQL query to create event - resetCredits

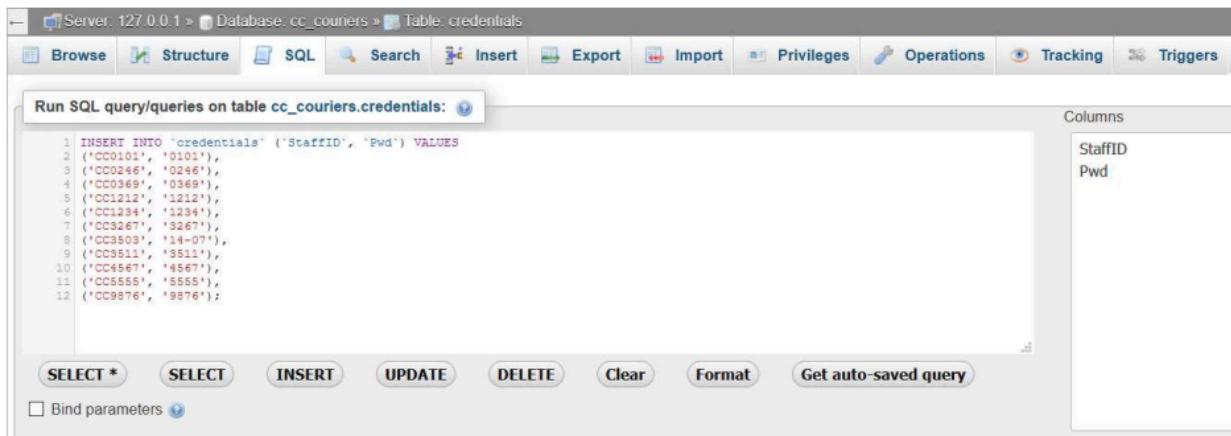
Name	Status	Action	Type
<input type="checkbox"/> resetCredits	ENABLED	Edit Export Drop	RECURRING

Check all With selected: Export Drop

Fig 7.2 : Result of successful creation of the event

5. INSERTING VALUES INTO THE TABLES :

Data is populated into credentials, staff, branches and pricing tables by the database administrator. Data in the remaining tables – parcel, status and feedback – are populated as the staff places and updates the parcel details and the customer files a feedback/query. 1. Credentials relation : Stores the login credentials (Staff Id and password) of each of the employees.

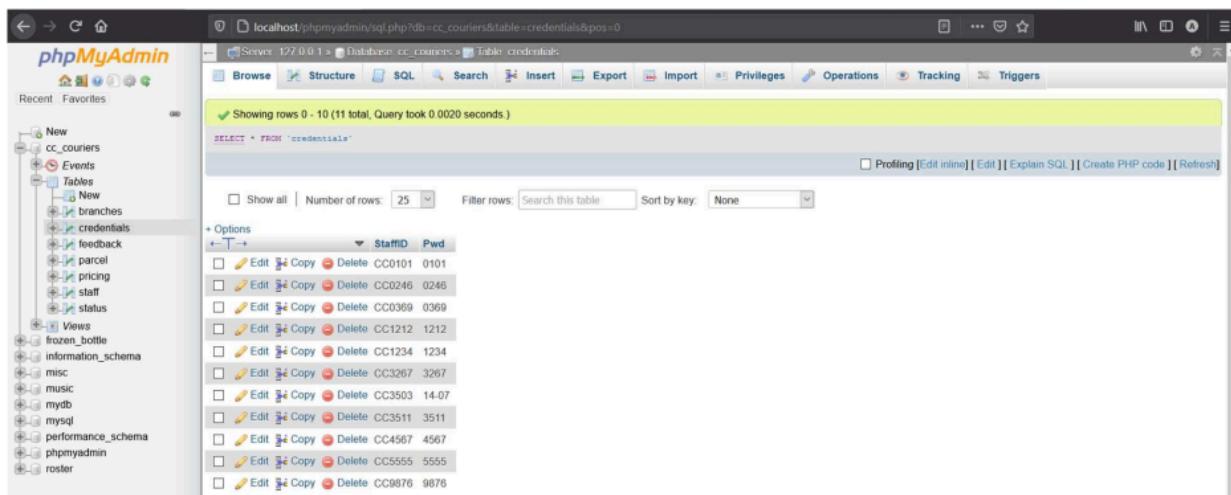


The screenshot shows the phpMyAdmin interface for the 'credentials' table. The SQL tab is active, displaying the following query:

```
1 INSERT INTO `credentials` ('StaffID', 'Pwd') VALUES
2 ('CC0101', '0101'),
3 ('CC0246', '0246'),
4 ('CC0369', '0369'),
5 ('CC1212', '1212'),
6 ('CC1234', '1234'),
7 ('CC3267', '3267'),
8 ('CC3503', '14-07'),
9 ('CC3511', '3511'),
10 ('CC4567', '4567'),
11 ('CC5555', '5555'),
12 ('CC9876', '9876');
```

Below the query, there are several buttons: SELECT *, SELECT, INSERT, UPDATE, DELETE, Clear, Format, Get auto-saved query, and Bind parameters. To the right, a 'Columns' panel lists 'StaffID' and 'Pwd'.

Fig 8.1 : SQL query to insert data into credentials table



The screenshot shows the phpMyAdmin interface for the 'credentials' table. The results tab is active, displaying the following data:

	StaffID	Pwd
<input type="checkbox"/>	CC0101	0101
<input type="checkbox"/>	CC0246	0246
<input type="checkbox"/>	CC0369	0369
<input type="checkbox"/>	CC1212	1212
<input type="checkbox"/>	CC1234	1234
<input type="checkbox"/>	CC3267	3267
<input type="checkbox"/>	CC3503	14-07
<input type="checkbox"/>	CC3511	3511
<input type="checkbox"/>	CC4567	4567
<input type="checkbox"/>	CC5555	5555
<input type="checkbox"/>	CC9876	9876

Fig 8.2 : Data stored in credentials table

2. Staff relation : Stores various details regarding the staff such as their id, name, salary, contact details, etc.

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** cc_couriers
- Table:** staff
- SQL Query:**

```
1 INSERT INTO `staff` ('StaffID', 'Name', 'Designation', 'Gender', 'DOB', 'DOJ', 'Salary', 'Mobile', 'Email', 'Credits') VALUES
2 ('CC3503', 'Stefan Salvatore', 'Branch Manager', 'Male', '1991-06-08', '2015-07-14', 55000, 1597324860, 'abc@xyz.com', 25),
3 ('CC9876', 'Arjun', 'Staff', 'Male', '1990-02-17', '2016-08-12', 35000, 9563287410, 'arjun@xyz.com', 0),
4 ('CC4567', 'Alaric Saltzman', 'Staff', 'Male', '1992-09-13', '2019-03-23', 38000, 8526459190, 'asaltzman@xyz.com', 10),
5 ('CC3511', 'Bonnie Bennet', 'Branch Manager', 'Female', '1998-07-24', '2019-04-09', 38000, 9487512306, 'bbennet@xyz.com', 0),
6 ('CC0246', 'Caroline Forbes', 'Branch Manager', 'Female', '1995-09-27', '2018-02-06', 40000, 9571368420, 'careforbes@xyz.com', 35),
7 ('CC1234', 'Hayley Marshal', 'Branch Manager', 'Female', '1996-07-12', '2017-06-08', 42000, 9658741230, 'haymarsh@xyz.com', 5),
8 ('CC5555', 'Josette', 'Staff', 'Female', '1989-05-09', '2019-05-15', 32000, 9856321470, 'jossie@xyz.com', 10),
9 ('CC0369', 'Kai Parker', 'Staff', 'Male', '1994-08-08', '2021-04-15', 25000, 9630258741, 'meetkai@xyz.com', 0),
10 ('CC0101', 'Millie B.', 'Staff', 'Female', '1995-03-12', '2020-12-12', 30000, 9571234560, 'milliebrown@xyz.com', 0),
11 ('CC1212', 'Klaus Mikelson', 'Branch Manager', 'Male', '1985-07-23', '2012-09-16', 55000, 9514782036, 'niklausmike@xyz.com', 5),
12 ('CC3267', 'Tyler Lockwood', 'Staff', 'Male', '2000-05-13', '2021-05-03', 23000, 9546287130, 'tylerwood@xyz.com', 20);
```
- Buttons:** SELECT*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, Get auto-saved query
- Columns:** StaffID, Name, Designation, Gender, DOB, DOJ, Salary, Mobile, Email, Credits

Fig 8.3 : SQL query to insert data into staff table

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** localhost
- Database:** cc_couriers
- Table:** staff
- Data View:** Showing rows 0 - 10 (11 total, Query took 0.0039 seconds.) [StaffID: CC0101... - CC9876...]
- SQL Query:**

```
SELECT * FROM `staff` ORDER BY `StaffID` ASC
```
- Table Headers:** StaffID, Name, Designation, Gender, DOB, DOJ, Salary, Mobile, Email, Credits
- Table Data:**

StaffID	Name	Designation	Gender	DOB	DOJ	Salary	Mobile	Email	Credits
CC0101	Millie B.	Staff	Female	1998-03-12	2012-09-16	30000	9871234560	milliebrown@xyz.com	0
CC0246	Caroline Forbes	Branch Manager	Female	1995-09-27	2018-02-06	40000	9571368420	careforbes@xyz.com	35
CC0369	Kai Parker	Staff	Male	1994-08-08	2021-04-15	25000	9630258741	meetkai@xyz.com	0
CC1212	Klaus Mikelson	Branch Manager	Male	1985-07-23	2012-09-16	55000	9514782036	niklausmike@xyz.com	5
CC3267	Hayley Marshal	Branch Manager	Female	1996-07-12	2017-06-08	42000	9658741230	haymarsh@xyz.com	5
CC3503	Tyler Lockwood	Staff	Male	2000-05-13	2021-05-03	23000	9546287130	tylerwood@xyz.com	20
CC3511	Stefan Salvatore	Branch Manager	Male	1991-06-08	2015-07-14	55000	1597324860	abc@xyz.com	25
CC4567	Bonnie Bennet	Branch Manager	Female	1998-07-24	2019-04-09	38000	9487512306	bbennet@xyz.com	0
CC5555	Josette	Staff	Female	1989-05-09	2019-05-15	32000	9856321470	jossie@xyz.com	10
CC9876	Arjun	Staff	Male	1990-02-17	2016-08-12	35000	9563287410	arjun@xyz.com	0

Fig 8.4 : Data stored in staff table

3. Branches relation : Stores the details of the various branch offices such as branch id, address, contact, email along with the manager id of that branch.

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** cc_couriers
- Table:** branches
- SQL Query:**

```
1 INSERT INTO `branches` ('Branch_id', 'Address', 'Contact', 'Email', 'Manager_id') VALUES
2 (1, '11, St.Joseph Church Campus, Dindigul - 624001', '4512424892', 'cccourierdgl@xyz.com', 'CC3511'),
3 (2, 'B-211, 11th Avenue, Ashok Nagar, Chennai - 600085', '443147782', 'cccourierchennai@xyz.com', 'CC0246'),
4 (3, '5E, Municipal Office Street, Tiruppur - 641604', '4214956251', 'cccourierstiruppur@xyz.com', 'CC1234'),
5 (4, '72, Arun complex, Brough Road, Erode - 638001', '4245276485', 'cccourierserode@xyz.com', 'CC1212'),
6 (5, '207, Abirami Nagar, Mettupalayam Road, Koundapalayam, Coimbatore - 641030', '3563343593', 'cccourierscbe@xyz.com', 'CC3503');
```
- Buttons:** SELECT*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, Get auto-saved query
- Columns:** Branch_id, Address, Contact, Email, Manager_id

Fig 8.5 : SQL query to insert data into branches table

Showing rows 0 - 4 (5 total, Query took 0.0022 seconds)

	Branch_Id	Address	Contact	Email	Manager_Id
<input type="checkbox"/>	1	11, St.Joseph Church Campus, Dindigul - 624001	4512424892	ccourierdgl@xyz.com	CC3511
<input type="checkbox"/>	2	B-211, 11th Avenue, Ashok Nagar, Chennai - 600083	4443147782	ccourierschennai@xyz.com	CC0246
<input type="checkbox"/>	3	5E, Municipal Office Street, Tiruppur - 641604	4214956251	ccouriersfiruppur@xyz.com	CC1234
<input type="checkbox"/>	4	72, Arun complex, Brough Road, Erode - 638001	4245276485	ccourierserode@xyz.com	CC1212
<input type="checkbox"/>	5	207, Abirami Nagar, Mettupalayam Road, Koundapalay...	3563343593	ccourierscbe@xyz.com	CC3503

Fig 8.6 : Data stored in branches table

4. Pricing relation : Stores the details of delivery charge per kg (as Cost) of the transportation between the corresponding states. Also the states present in this relation depicts the availability of the service in various states mentioned in it.

```

1 INSERT INTO `pricing` ('S.No', `State_1`, `State_2`, `Cost`) VALUES
2 (0, 'Tamil Nadu', 'Tamil Nadu', 50),
3 (1, 'Tamil Nadu', 'Kerala', 100),
4 (2, 'Tamil Nadu', 'Andhra Pradesh', 125),
5 (3, 'Tamil Nadu', 'Karnataka', 130),
6 (4, 'Tamil Nadu', 'Maharashtra', 200),
7 (5, 'Tamil Nadu', 'Delhi', 300),
8 (6, 'Tamil Nadu', 'Uttar Pradesh', 350),
9 (7, 'Tamil Nadu', 'West Bengal', 275);
10

```

Fig 8.7 : SQL query to insert data into pricing table

Showing rows 0 - 7 (8 total, Query took 0.0019 seconds.)

	S.No	State_1	State_2	Cost
<input type="checkbox"/>	0	Tamil Nadu	Tamil Nadu	50
<input type="checkbox"/>	1	Tamil Nadu	Kerala	100
<input type="checkbox"/>	2	Tamil Nadu	Andhra Pradesh	125
<input type="checkbox"/>	3	Tamil Nadu	Karnataka	130
<input type="checkbox"/>	4	Tamil Nadu	Maharashtra	200
<input type="checkbox"/>	5	Tamil Nadu	Delhi	300
<input type="checkbox"/>	6	Tamil Nadu	Uttar Pradesh	350
<input type="checkbox"/>	7	Tamil Nadu	West Bengal	275

Fig 8.8 : Data stored in pricing table

Sample Screen shots

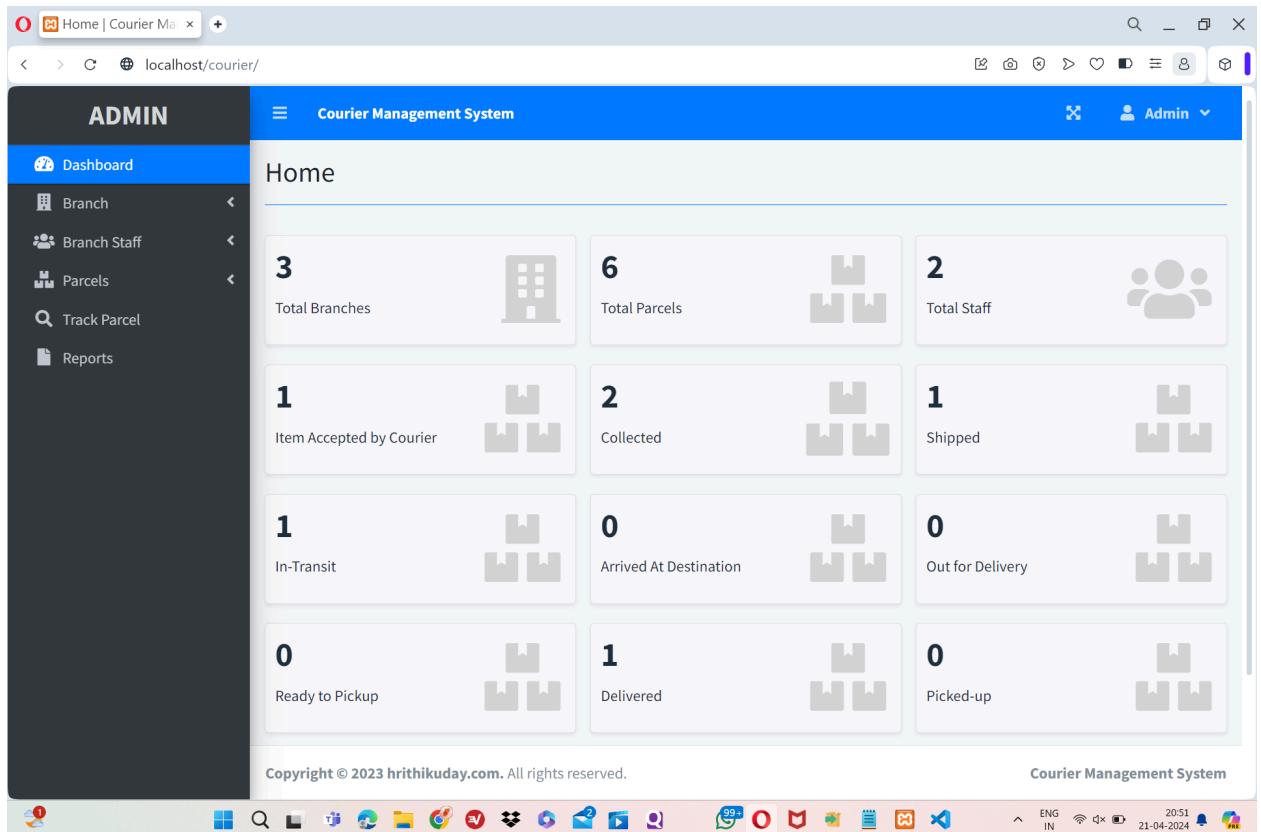


Fig 9: Screengrab of the website

The program module specification describes the specification of what the program would do to enhance good design guide during the process. As it concerns this particular online courier management system, it describes different activities involved in the program modules. The classifications of the modules are described below. There are total of four modules in this system specified according to their functions. They include:

- 1. Administrators Login Module:** This area enables the administrator to login into the admin area either to add, delete, update or make any adjustments to the database of the system.
- 2. Customer Delivery Module:** The customer Delivery Modules helps the customer to track the progress of his or her delivery and also to check the delivery status of their parcels.
- 3) Branch Management Branch** who provide courier service Not for direct to customer it goes through a branch . It is a one type of small module which is include in any type of Courier System and also our web application content is able to use for any other Online Service Management. Where they wants to add any distributor of their company on their system to register themselves Online Service Management System.

The screenshot shows a web application titled "Courier Management System". On the left, there is a sidebar with a dark theme and white text. It has a header "ADMIN" and a main menu with "Parcels" selected. Under "Parcels", there are several options: "Add New", "List All", and several status-based filters: "Item Accepted by Courier", "Collected", "Shipped", "In-Transit", "Arrived At Destination", "Out for Delivery", "Ready to Pickup", "Delivered", "Picked-up", and "Unsuccessfull".

The main content area displays a table of parcel data. The columns are: #, Reference Number, Sender Name, Recipient Name, Status, and Action. There are 6 entries listed:

#	Reference Number	Sender Name	Recipient Name	Status	Action
1	002940909648	Suryansh Singh	VARTIKA	Collected	
2	505604168988	Suryansh Singh	Hrithik	In-Transit	
3	514912669061	UDAY	VARTIKA	Item Accepted by Courier	
4	117967400213	SHIVAM	PRANAV	Collected	
5	983186540795	FARHAN	UDAY	Shipped	
6	201406231415	BHAVYA	PARUL	Delivered	

At the bottom of the table, it says "Showing 1 to 6 of 6 entries". Below the table, there is a copyright notice: "Copyright © 2023 hrithikuday.com. All rights reserved." and the text "Courier Management System".

Fig 10: Screengrab of the website

CONNECTING TO DATABASE :

```
db_connect.php
1 <?php
2 // connect to the database
3 $conn = mysqli_connect('localhost', 'Torvus', 'Aji1407', 'cc_couriers');
4 // check connection
5 if(!$conn){
6     echo 'Connection error: '. mysqli_connect_error();
7 }
8 ?>
```

Fig 10.1 : Code for connecting the database

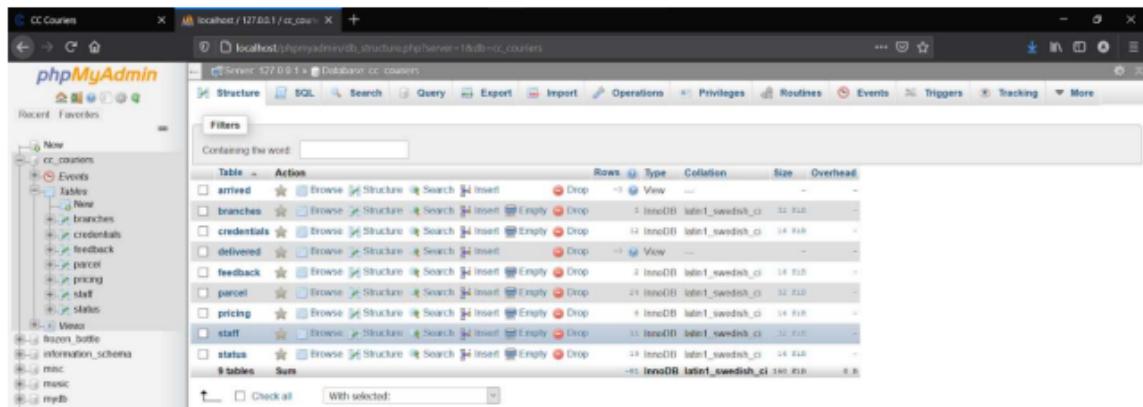


Fig 10.2 : CC Couriers Database

STAFF LOGIN :

```

login.php
1 <?php
2     include("db_connect.php");
3
4     $id = $pwd = '';
5     $errors = array('id' => '', 'pwd' => '', 'login' => '');
6
7     if(isset($_POST['submit'])){
8         if(empty($_POST['id'])){
9             $errors['id'] = "*Required";
10        }else{
11            $id = $_POST['id'];
12        }
13        if(empty($_POST["pwd"])){
14            $errors['pwd'] = "*Required";
15        }else{
16            $pwd = $_POST['pwd'];
17        }
18        if(array_filter($errors)){
19            //echo errors
20        }else{
21            $id = mysqli_real_escape_string($conn, $id);
22            $pwd = mysqli_real_escape_string($conn, $pwd);
23
24            $sql = "SELECT * FROM credentials WHERE StaffID='$id' AND Pwd='$pwd'";
25            $result = mysqli_query($conn, $sql);
26            if(mysqli_num_rows($result) > 0){
27                $user = mysqli_fetch_assoc($result);
28                session_start();
29                $_SESSION['id'] = $user['StaffID'];
30                header("Location: staff.php");
31            }else{
32                $sql = "SELECT * FROM credentials WHERE StaffID='$id'";
33                $result = mysqli_query($conn, $sql);
34                if(mysqli_num_rows($result) == 0){
35                    $errors['login'] = 'Enter valid Staff ID';
36                }else{
37                    $user = mysqli_fetch_assoc($result);
38                    if($pwd != $user['Pwd']){
39                        $errors['login'] = 'Incorrect Password';
40                    }
41                }
42            }
43        }
44    }

```

Fig 11.1 : PHP code snippet for validating the login credentials and redirecting to the staff page on successful login. It stores the credentials in the session on successful login.

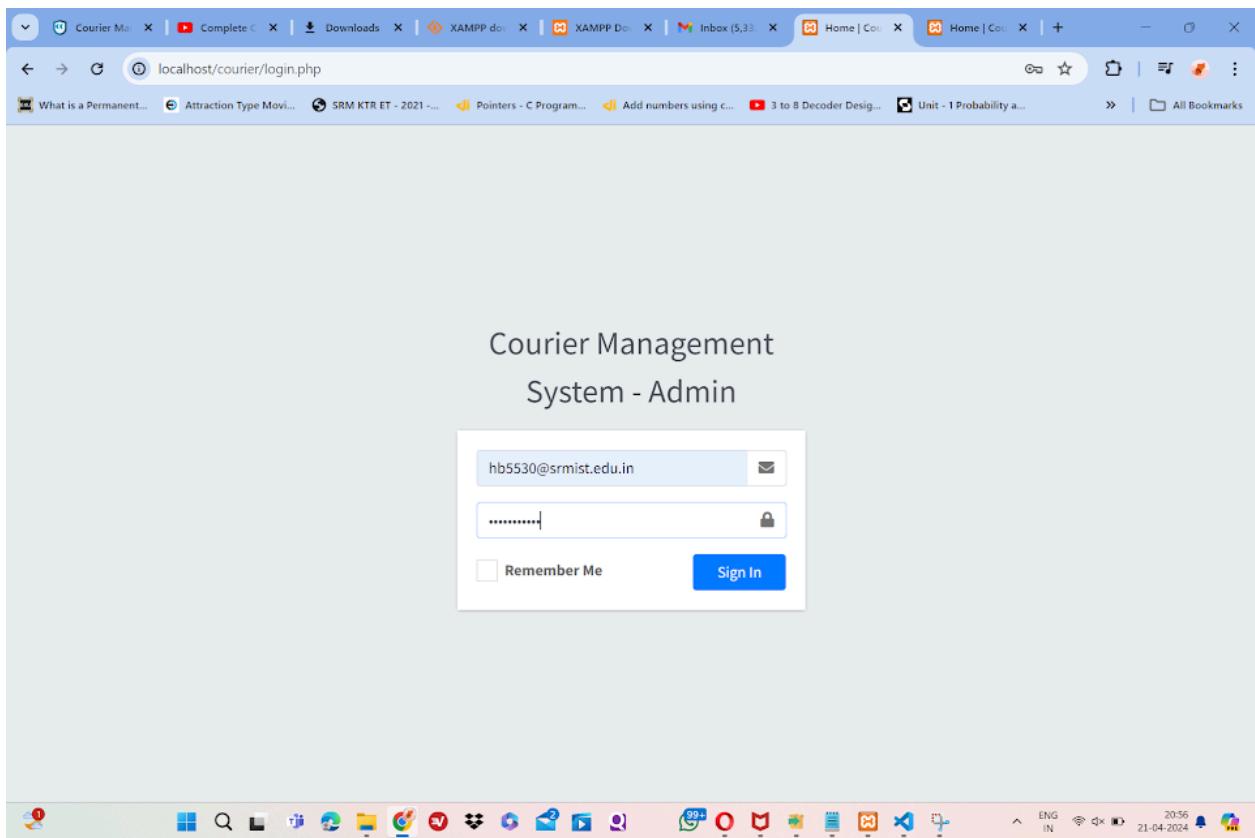


Fig 11.2 : Staff login page with login credentials as per the credentials table

#	Branch Code	Street/Building/Brgy.	City/State/Zip	Country	Contact #	Action
1	Kylab3mYBgAX71t	CP	New Delhi, New Delhi, 400023	India	+1234567489	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
2	dlbUK5mEh96f0Zc	Karol Bagh	New Delhi, New Delhi, 400087	India	9876543210	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
3	vzTLOPqMogyOWhF	Srm Nagar	Chengalpattu, Tamil Nadu, 603203	India	12345678	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>

The screenshot shows a web-based Courier Management System interface. The left sidebar, titled "ADMIN", contains links for Dashboard, Branch, Branch Staff, Parcels, and Track Parcel (which is currently selected). The main content area has a header "Courier Management System" and a sub-header "Track". A search bar at the top right allows entering a tracking number, with "201406231415" already input. Below the search bar is a timeline of seven delivery status updates:

- Item accepted by Courier (Nov 26, 2020 04:15 PM)
- Collected (Nov 27, 2020 10:28 AM)
- Shipped (Nov 27, 2020 10:28 AM)
- In-Transit (Nov 27, 2020 10:28 AM)
- Arrived At Destination (Nov 27, 2020 11:05 AM)
- Out for Delivery (Nov 27, 2020 11:05 AM)
- Delivered (Nov 27, 2020 11:05 AM)

At the bottom, a copyright notice reads "Copyright © 2023 hrithikuday.com. All rights reserved." and the system name is "Courier Management System".

The screenshot shows a web-based Courier Management System. The top navigation bar includes a logo, a search bar, and user authentication for 'Admin'. A left sidebar titled 'ADMIN' lists several menu items: Dashboard, Branch, Branch Staff, Parcels, Track Parcel, and Reports, with 'Reports' currently selected. The main content area is titled 'Courier Management System' and displays a 'Reports' section. This section features a search bar with fields for 'Status' (set to 'All'), 'From' (06-06-2019), 'To' (21-04-2024), and a 'View Report' button. Below the search bar is a table listing six courier transactions. The table columns are: #, Date, Sender, Recipient, Amount, and Status. The data is as follows:

#	Date	Sender	Recipient	Amount	Status
1	Nov 26, 2020	BHAVYA	PARUL	2,500.00	Delivered
2	Nov 26, 2020	SHIVAM	PRANAV	2,500.00	Collected
3	Nov 26, 2020	FARHAN	UDAY	1,500.00	Shipped
4	Nov 27, 2020	UDAY	VARTIKA	1,900.00	Item Accepted by Courier
5	Nov 27, 2020	Suryansh Singh	Hrithik	2,500.00	In-Transit
6	Apr 21, 2024	Suryansh Singh	VARTIKA	10,678.00	Collected

At the bottom of the page, there is a copyright notice for hrithikuday.com and a footer bar with various system icons.

Staff List

#	Staff	Email	Branch	Action
1	Hrithik H	hb5530@srmist.edu.in	Srm Nagar, Chengalpattu, Tamil Nadu, 603203, India	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
2	Uday Singh	uh4674@srmist.edu.in	Karol Bagh, New Delhi, New Delhi, 400087, India	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>

Showing 1 to 2 of 2 entries

Copyright © 2023 hrithikuday.com. All rights reserved. Courier Management System

Fig 12.2 : Page displaying the fetched staff details from the staff table

4/21/24, 8:56 PM

about:blank

Report

Date Range: 2019-06-06 to 2024-04-21

Status: All

#	Date	Sender	Recipient	Amount	Status
1	Nov 26, 2020	BHAVYA	PARUL	2,500.00	Delivered
2	Nov 26, 2020	SHIVAM	PRANAV	2,500.00	Collected
3	Nov 26, 2020	FARHAN	UDAY	1,500.00	Shipped
4	Nov 27, 2020	UDAY	VARTIKA	1,900.00	Item Accepted by Courier
5	Nov 27, 2020	Suryansh Singh	Hrithik	2,500.00	In-Transit
6	Apr 21, 2024	Suryansh Singh	VARTIKA	10,678.00	Collected

Conclusion

A courier delivers messages, packages, and mail. Couriers are distinguished from ordinary mail services by features such as speed, security, tracking, signature, specialization and individualization of express services, and swift delivery times, which are optional for most every day mail services. As a premium service, couriers are usually more expensive than standard mail services, and their use is typically restricted to packages where one or more of these features are considered important enough to warrant the cost. The aim of this research project is to design and implement a Courier Service Packaging and Delivery Management System that will automate the process of delivery tracking and monitoring for the recipients of the deliveries.

From understanding the user and admin requirements to system design and finally consolidation of everything, each step requires in-depth understanding and commitment toward achieving the objectives of this project. Couriers is a customer-friendly delivery service with flexible policies and proper management system. Although the Courier Management System developed in this project is not fully integrated to the real world needs of a system, the prototype and implementation demonstrates easy navigation in the system and how data are stored in a systematic view. The study of how a relational database will be developed and maintained for industry purposes was done and been implemented. Overall, the main motive of this project is to gain more knowledge about the usage of database systems in the industrial view which was done efficiently. Also it is indeed a great learning experience.

References

- <https://www.w3schools.com/>
- <https://www.javatpoint.com/dbms-tutorial>
- <https://www.guru99.com>
- www.stackoverflow.com
- <https://www.youtube.com/>
- <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- **Database System Concepts, Seventh Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan**