DATABASE MANAGEMENT SYSTEM

ASSIGNMENT- 4

UE19CS301

Team - 24

Divyanshu Sharma: PES1UG20CS806

Navya C: PES1UG20CS818

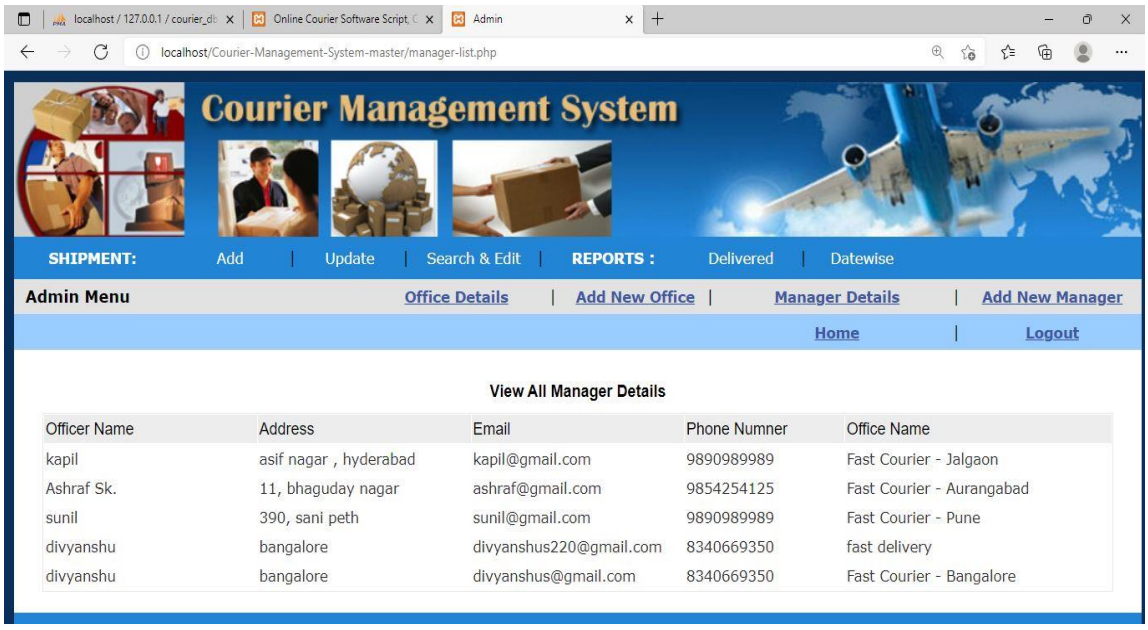
Dependencies installed for the database connectivity:

The front-end of our database was implemented using PHP , which is a popular front-end framework which uses CSS and JavaScript-based design templates.

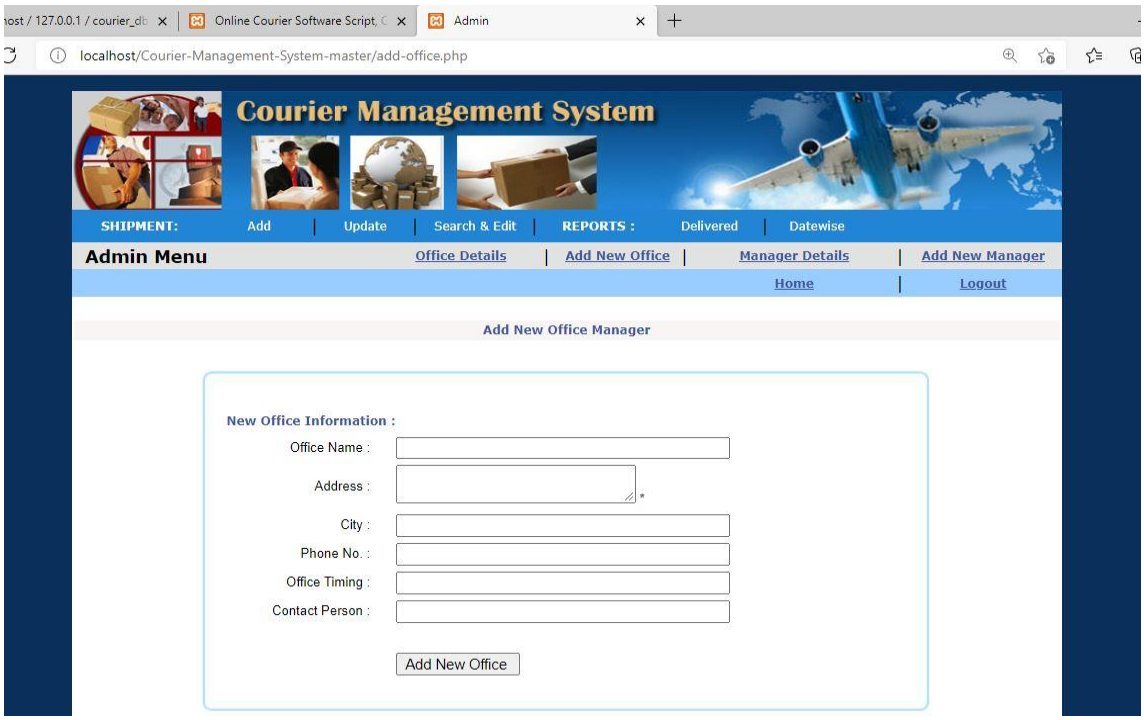
We utilized MYSQL to structure our database and the required dependencies used to connect the front-end to the back-end was phpMyAdmin, which is a free and open-source administration tool used for MYSQL .

Execution on the frontend with sample queries:

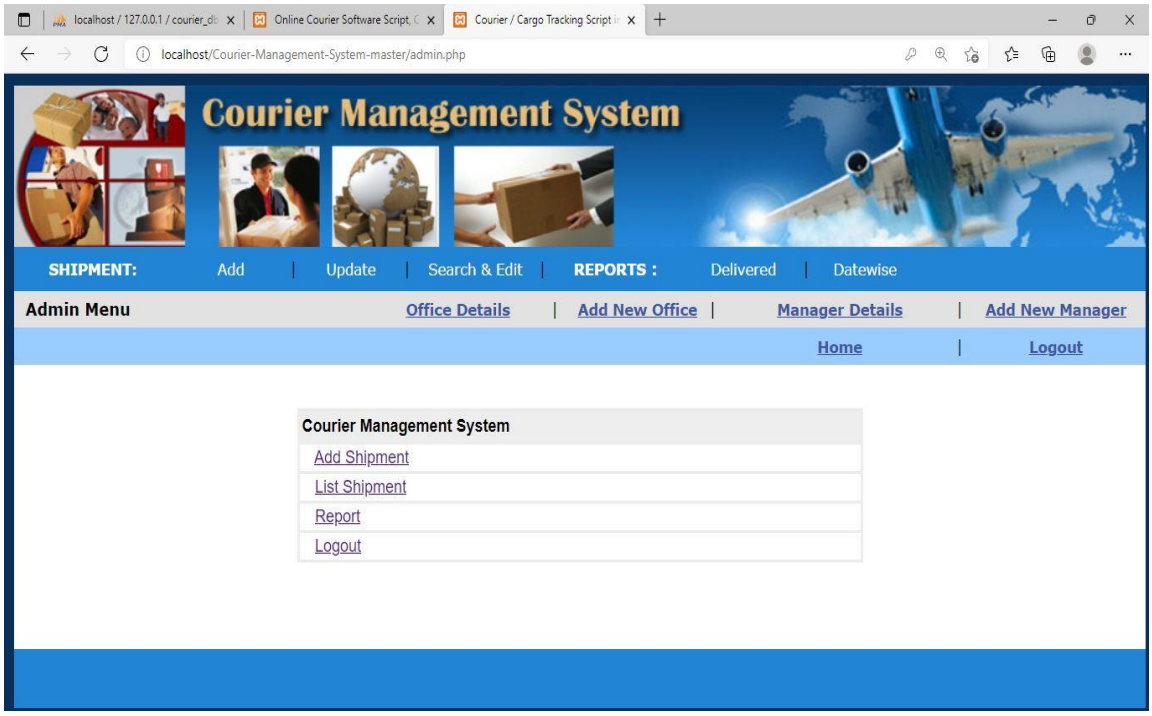
Viewing all Manager Details:



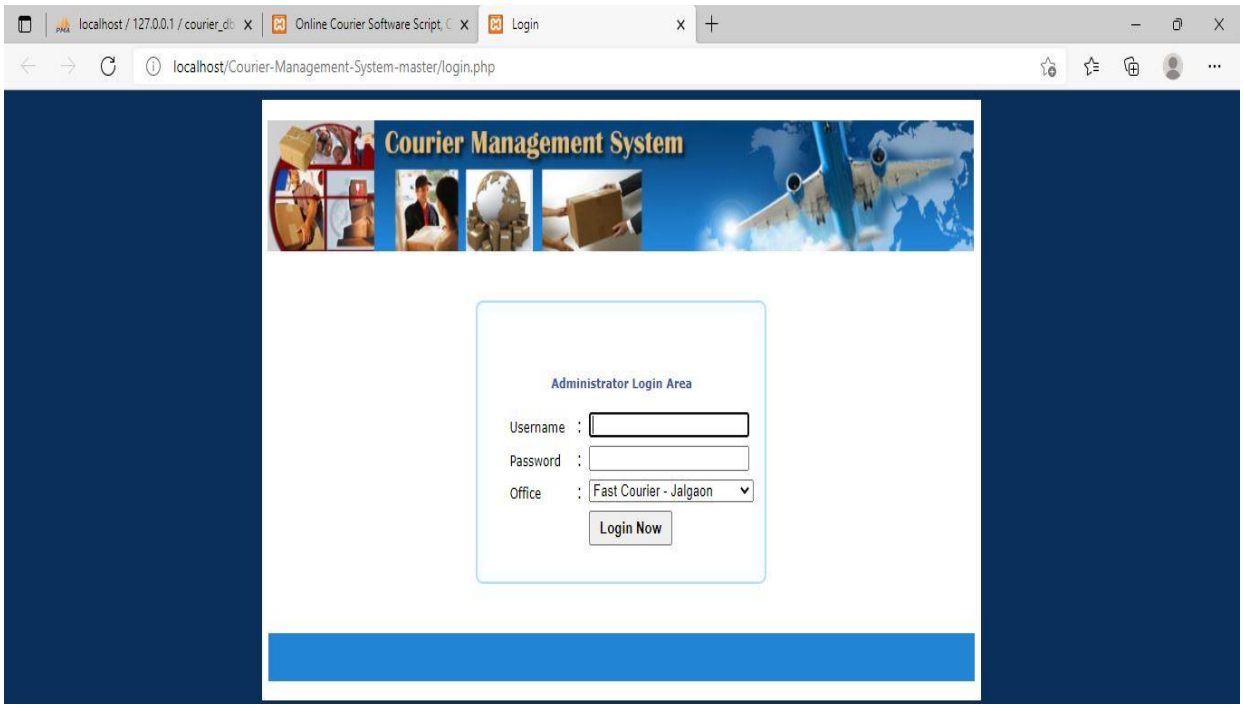
Adding New Office Manager Details:



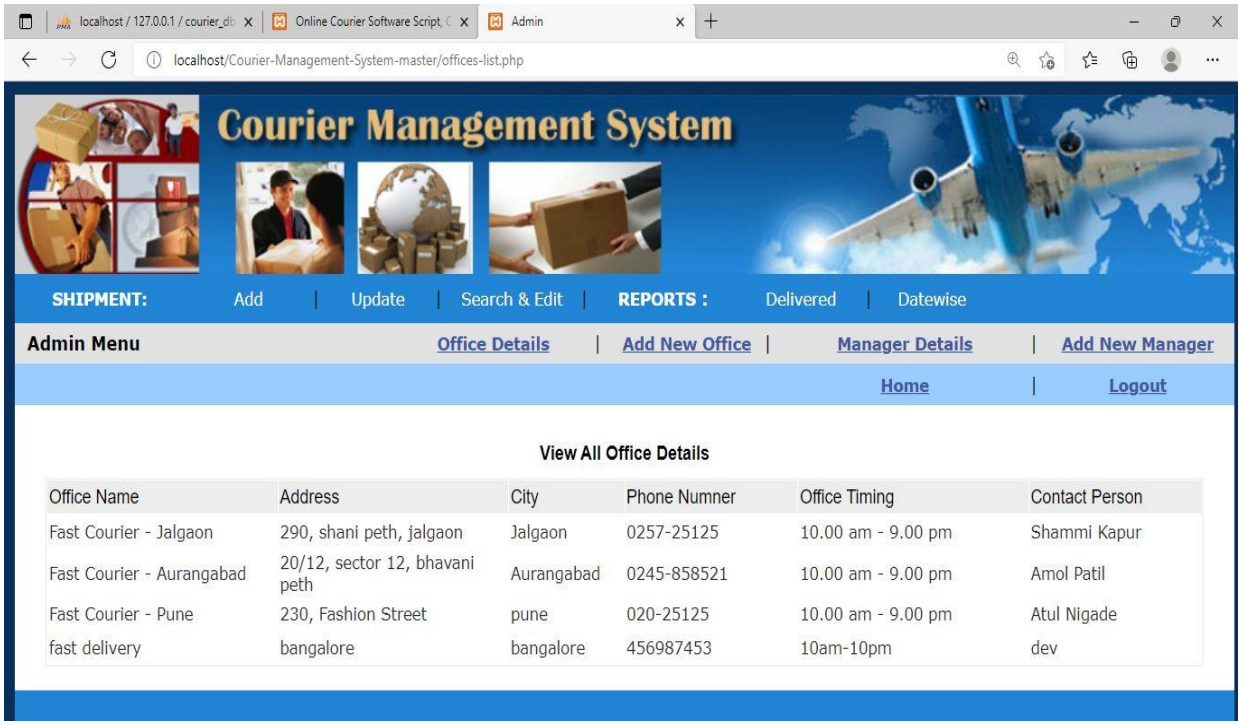
Home Page:



Administrator Login Page:

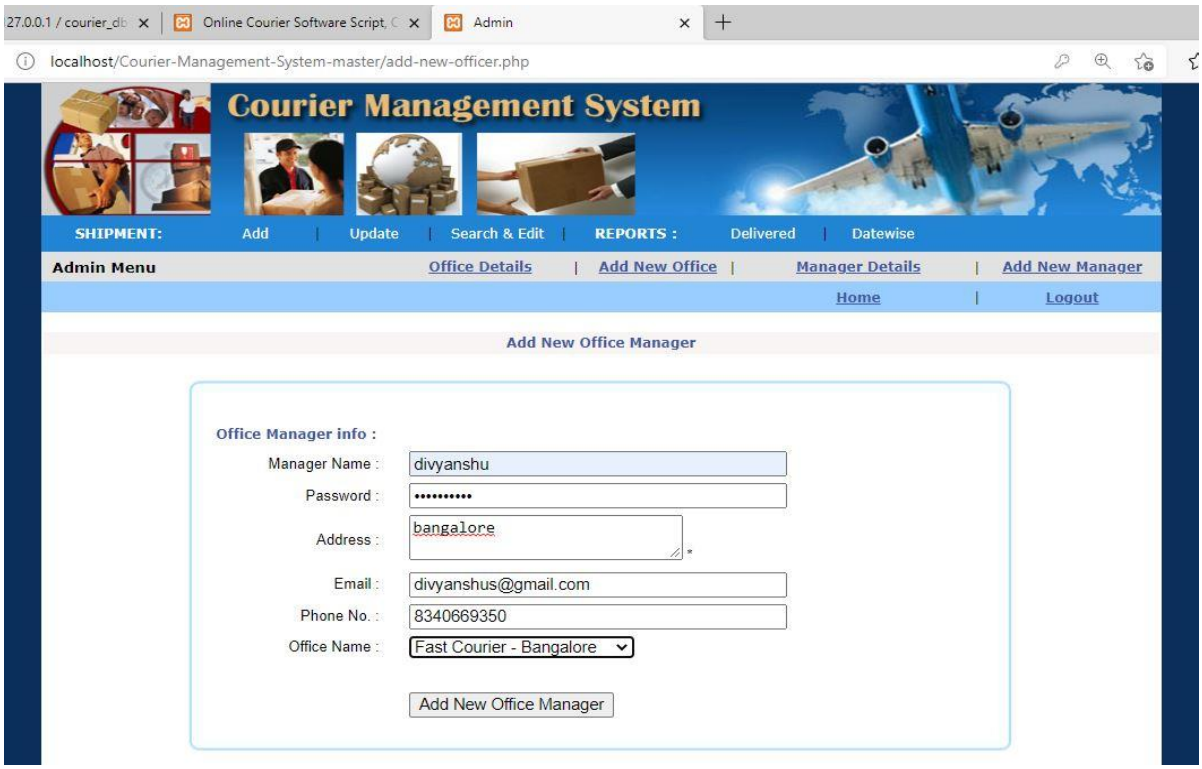


Viewing all Office Details:

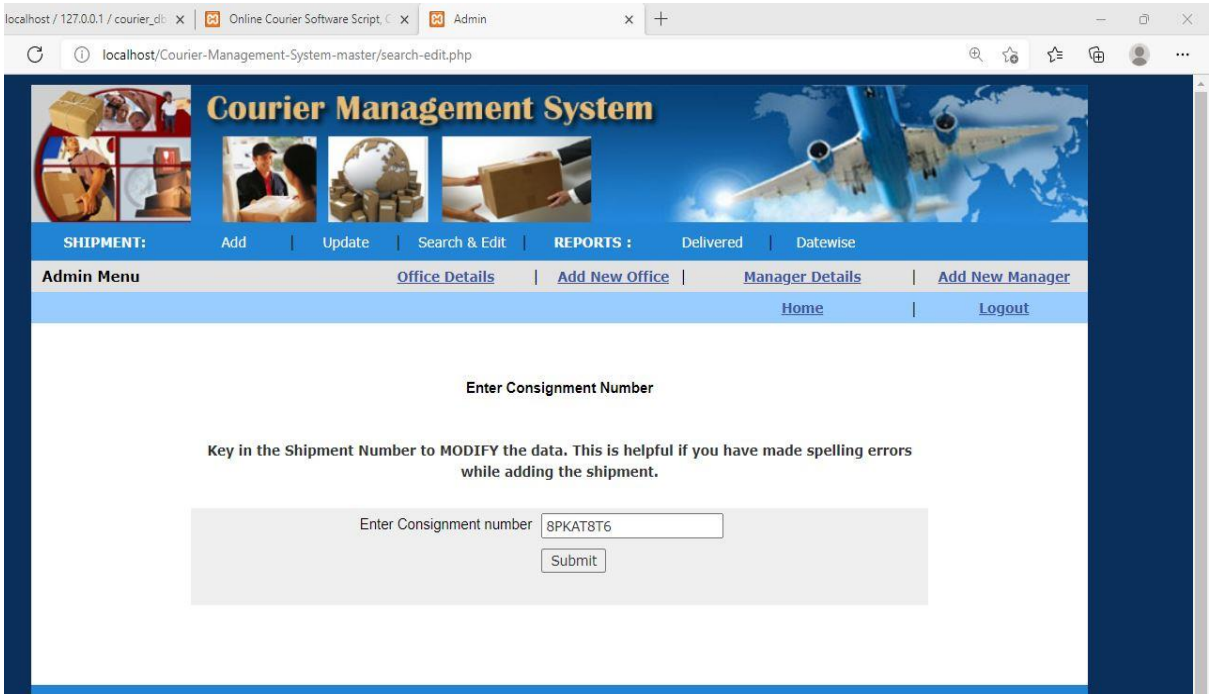


Office Name	Address	City	Phone Numner	Office Timing	Contact Person
Fast Courier - Jalgaon	290, shani peth, jalgaon	Jalgaon	0257-25125	10.00 am - 9.00 pm	Shammi Kapur
Fast Courier - Aurangabad	20/12, sector 12, bhavani peth	Aurangabad	0245-858521	10.00 am - 9.00 pm	Amol Patil
Fast Courier - Pune	230, Fashion Street	pune	020-25125	10.00 am - 9.00 pm	Atul Nigade
fast delivery	bangalore	bangalore	456987453	10am-10pm	dev

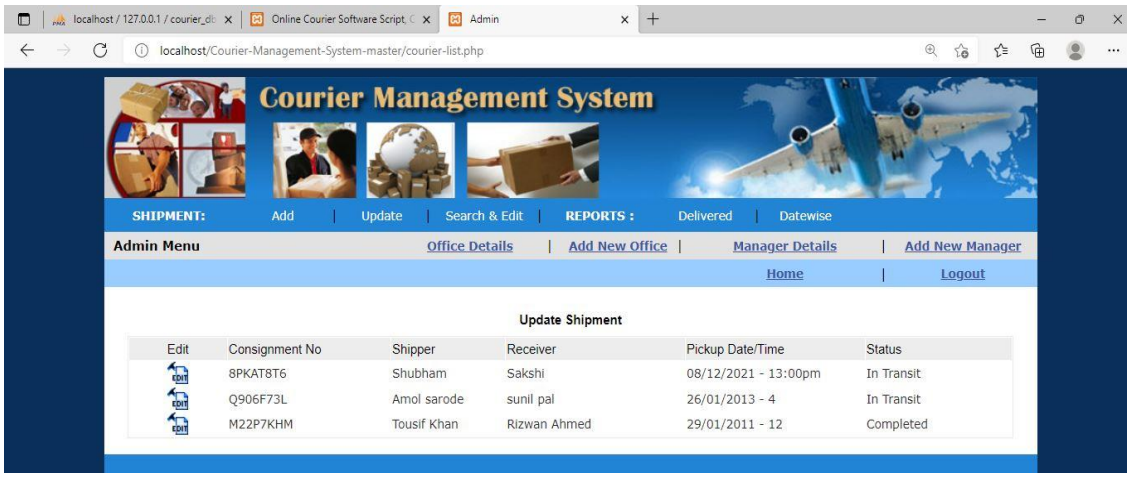
Adding New Office Manager Details:

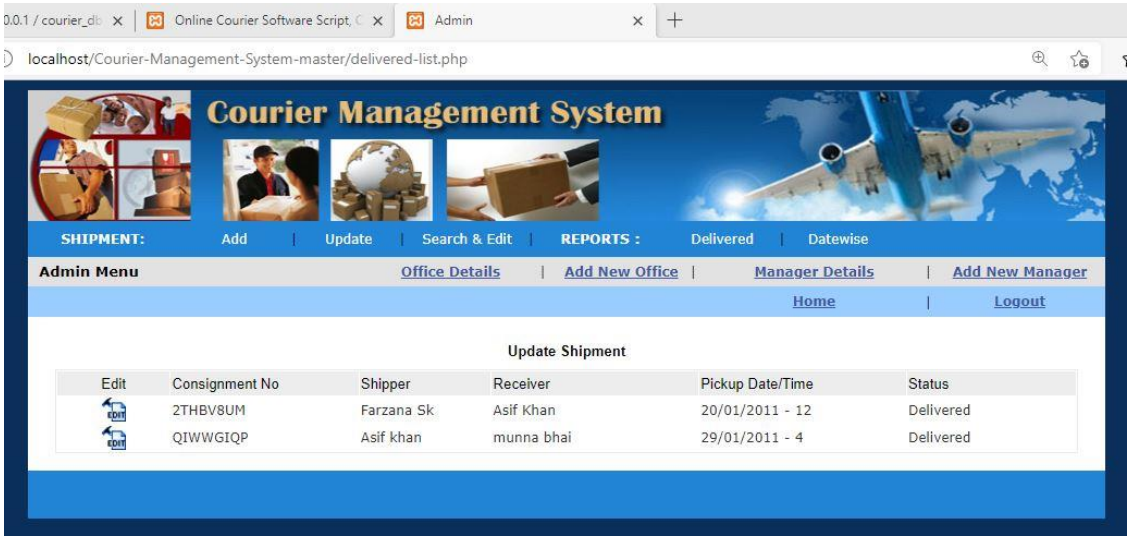


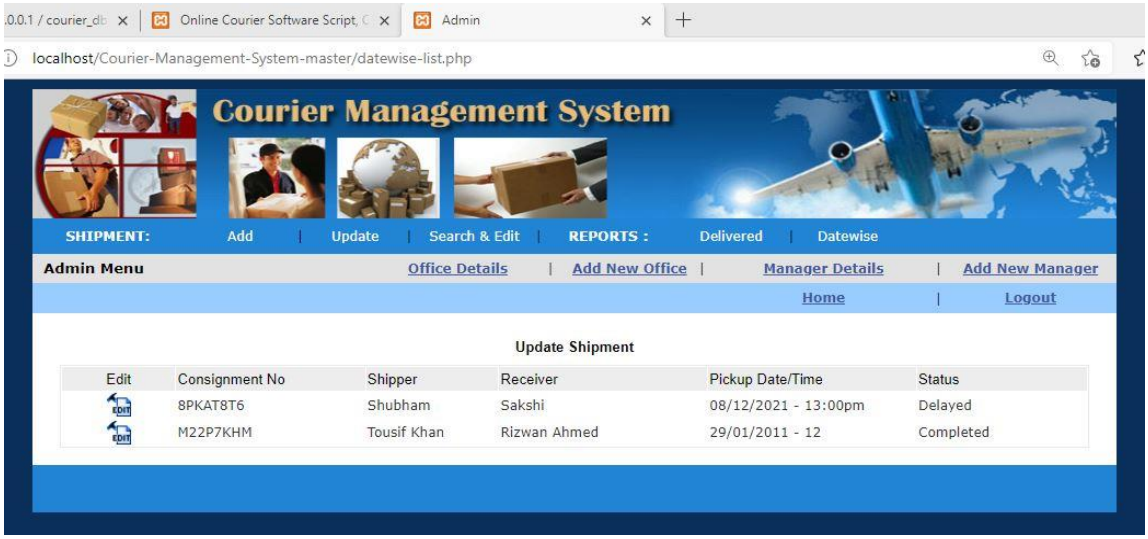
Entering Consignment Number:



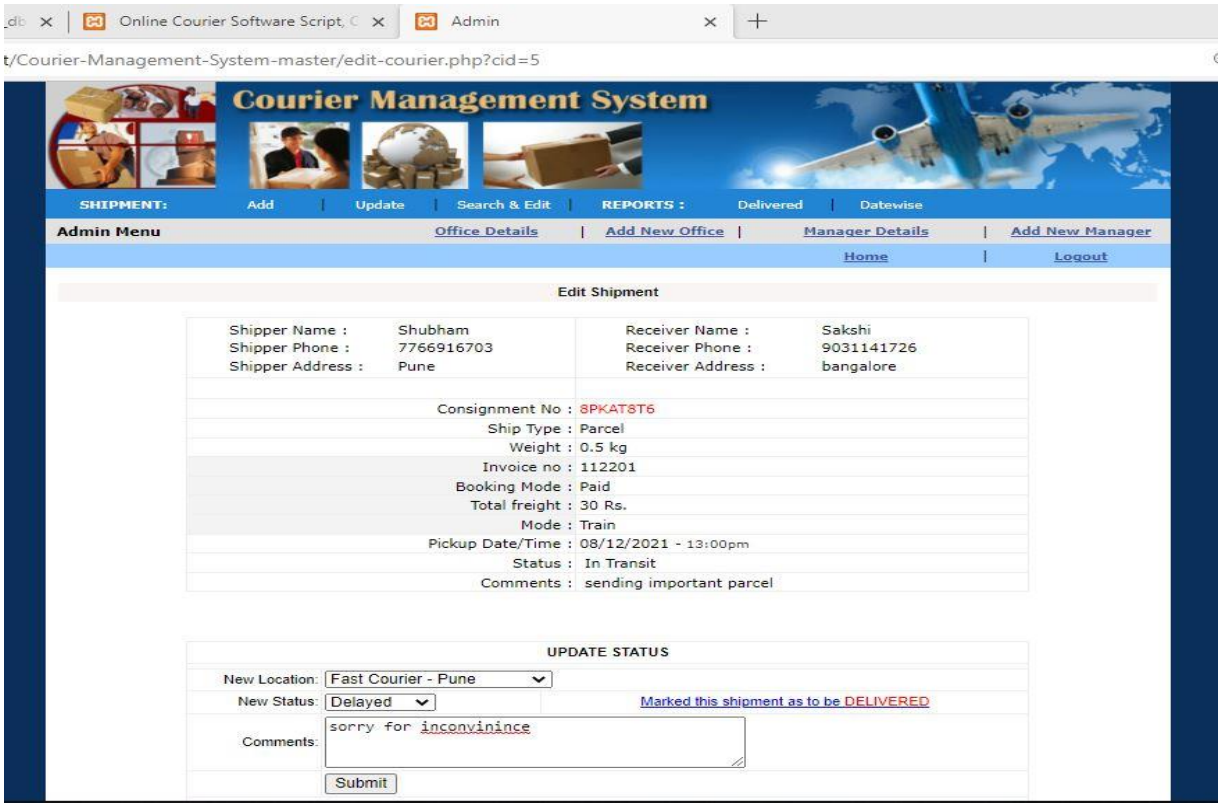
Updating Shipment Details:



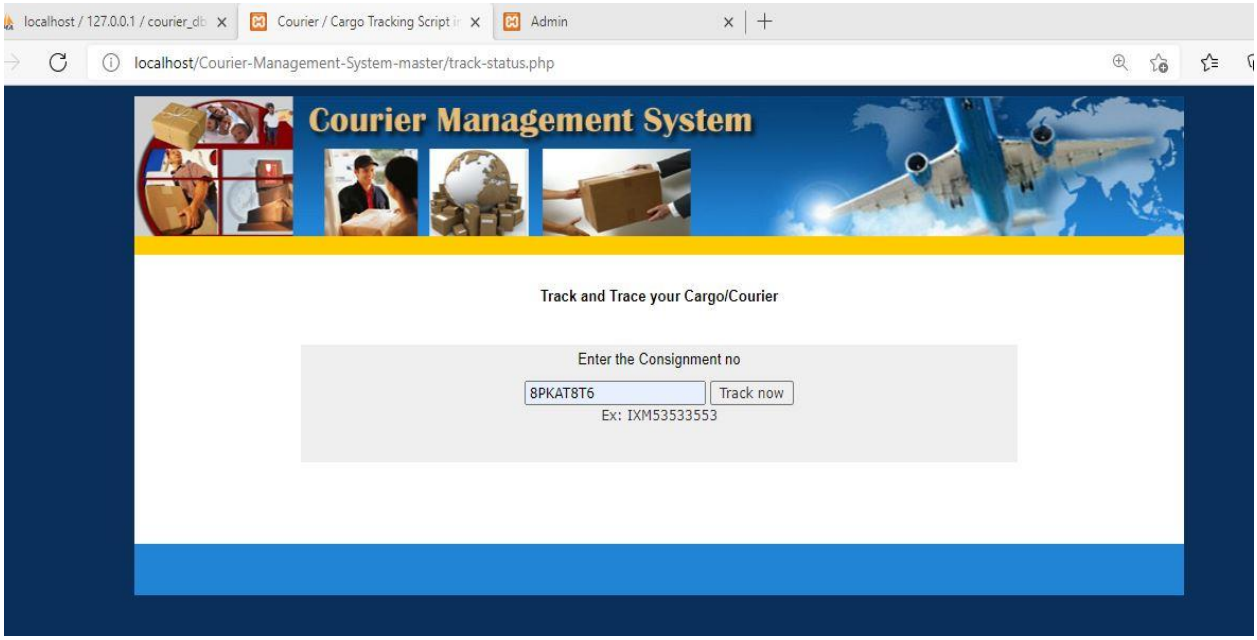




Editing Shipment Details:



Tracking and Tracing the Courier:



Schema & Constraint Change Statement

- For the ‘couriers’ entity: **Source, Destination** and **dest_ph** attributes were added.

ALTER TABLE couriers ADD Source varchar(50);

```
courier=# ALTER TABLE couriers ADD Source varchar(50);
ALTER TABLE
courier=# \c couriers;
FATAL:  database "couriers" does not exist
Previous connection kept
courier=# \d couriers;
                    Table "public.couriers"
    Column    |         Type          | Collation | Nullable | Default
--------------+-----------------------+-----------+----------+---------
 cour_id      | integer               |           | not null |
 shipment_id  | integer               |           | not null |
 office_id    | integer               |           | not null |
 courier_name | character varying(40) |           | not null |
 destination  | character varying(50) |           |          |
 dest_ph      | character varying(12) |           |          |
 source       | character varying(50) |           |          |
Indexes:
    "couriers_pkey" PRIMARY KEY, btree (cour_id, courier_name)
```

ALTER TABLE couriers ADD Destination varchar(50);

```
courier=# ALTER TABLE couriers ADD destination varchar(50);
ALTER TABLE
courier=# \d couriers;
                    Table "public.couriers"
    Column    |         Type          | Collation | Nullable | Default
--------------+-----------------------+-----------+----------+---------
 cour_id      | integer               |           | not null |
 shipment_id  | integer               |           | not null |
 office_id    | integer               |           | not null |
 courier_name | character varying(40) |           | not null |
 dest_ph      | character varying(12) |           |          |
 source       | character varying(50) |           |          |
 destination  | character varying(50) |           |          |
Indexes:
    "couriers_pkey" PRIMARY KEY, btree (cour_id, courier_name)
```

ALTER TABLE couriers ADD dest_ph varchar(12);

```
courier=# ALTER TABLE couriers ADD dest_ph varchar(50);
ALTER TABLE
courier=# \d couriers;
                    Table "public.couriers"
    Column    |         Type          | Collation | Nullable | Default
--------------+-----------------------+-----------+----------+---------
 cour_id      | integer               |           | not null |
 shipment_id  | integer               |           | not null |
 office_id    | integer               |           | not null |
 courier_name | character varying(40) |           | not null |
 source       | character varying(50) |           |          |
 destination  | character varying(50) |           |          |
 dest_ph      | character varying(50) |           |          |
Indexes:
    "couriers_pkey" PRIMARY KEY, btree (cour_id, courier_name)
```

- For the ‘customer’ entity: **password** attribute was added

ALTER TABLE customer ADD password varchar(255);

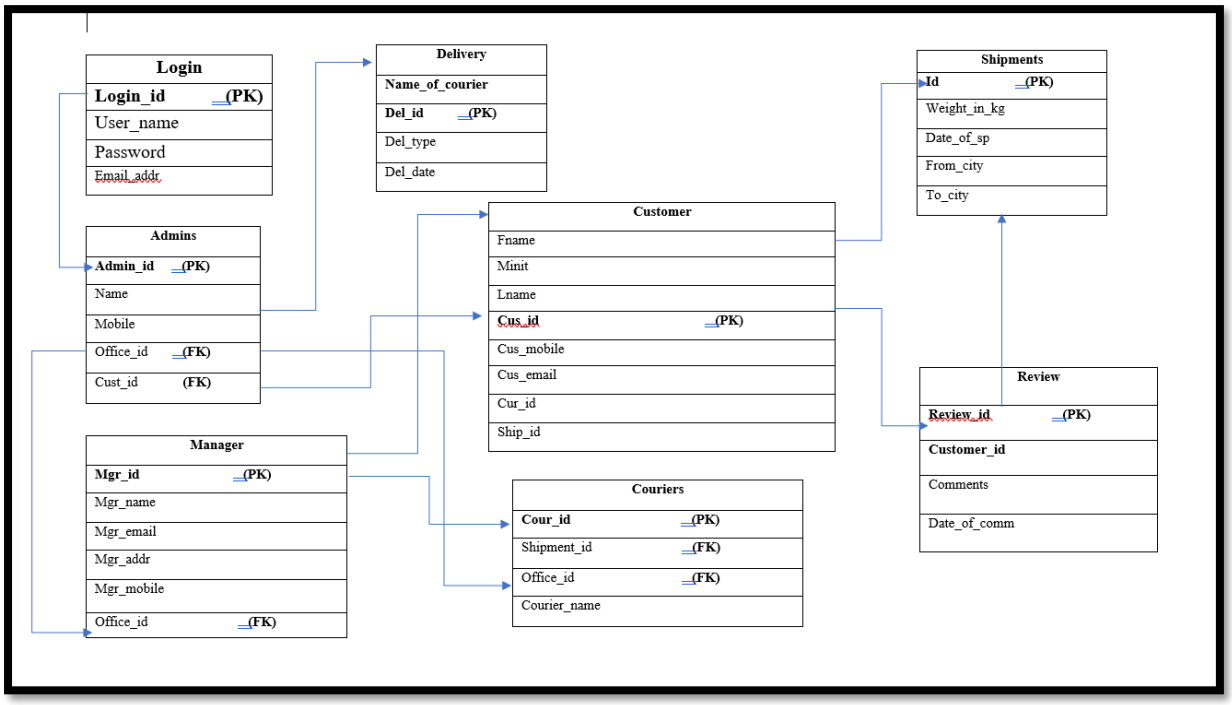
```
courier=# ALTER TABLE customer ADD password varchar(50);
ALTER TABLE
courier=# \d customer;
                      Table "public.customer"
   Column    |         Type          | Collation | Nullable | Default
-------------+-----------------------+-----------+----------+---------
 cus_id      | integer               |           | not null |
 fname       | character varying(30) |           | not null |
 minit       | character varying(20) |           |          |
 lname       | character varying(20) |           |          |
 cus_mobile  | integer               |           | not null |
 cus_email   | character varying(40) |           | not null |
 cur_id      | integer               |           | not null |
 ship_id     | integer               |           | not null |
 password    | character varying(50) |           |          |
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cus_id)
```

- For the ‘delivery’ entity: **custname** and **courier_id(foreign key)** attributes were added

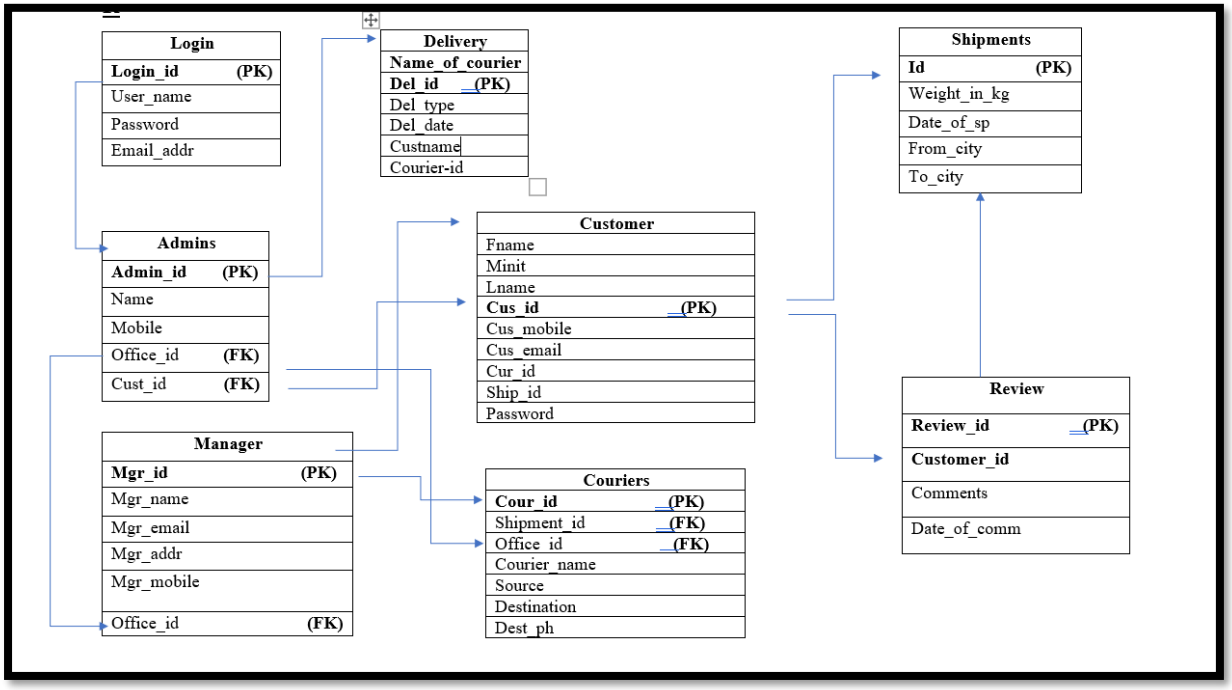
ALTER TABLE delivery ADD custname varchar(25);

```
courier=# ALTER TABLE delivery ADD custname varchar(50);
ALTER TABLE
courier=# \d delivery;
                      Table "public.delivery"
     Column      |         Type          | Collation | Nullable | Default
-----------------+-----------------------+-----------+----------+---------
 name_of_courier | character varying(50) |           |          |
 del_id          | integer               |           | not null |
 del_type        | character varying     |           | not null |
 del_date        | date                  |           |          |
 custname        | character varying(50) |           |          |
Indexes:
    "delivery_pkey" PRIMARY KEY, btree (del_id)
```

Before:



After:



Changes in Business/Application:

Changes in the Application or expansion is to make working software available at cloud, so that customers can access and process it at all time.

With the existing design of your database, if you have to migrate to any No-SQL variety, which one will be your choice? Why?

If we have to migrate to a No-SQL variety then our preferred choice would be a document database. These databases store data in CSV,JSON, BSON, or XML formats. These documents can be retrieved and stored in a similar format as that of data objects used in applications which implies minimal requirement for translations. Document databases are known to be very flexible i.e. structuring and reworking the databases are per need, which not only makes it time efficient but also allows its developers to have full control over it.

Migration from Mysql to Postgresql:
Link : enterprisedb.com/postgres-t-export-data-using-csv-files-postgresql

The most commonly used document database is POSTGRE SQL.

Following are a few key differences between MySQL and POSTGRE SQL:

- MySQL is a Relational Database Management System (RDBMS) and PostgreSQL is an object-relational Database Management System (ORDBMS). In Relational Database Models, the database is represented as a collection of relations. An ORDBMS has qualities of an RDBMS and in addition to that, it has several features of object-oriented management systems like objects, classes, and inheritance.
- PostgreSQL is ACID (Atomicity Consistency Isolation Durability) compliant while MySQL is not ACID-compliant. MySQL is ACID-compliant only with Inno DB (an ACID-compliant storage engine for MySQL) and NDB (Network Database) cluster engines.
- PostgreSQL has Multi-Version Concurrency Control (MVCC) which enables multiple users to work on a PostgreSQL database simultaneously. MySQL provides MVCC support only after using Inno DB.
- MySQL provides MySQL Workbench as a GUI tool, PostgreSQL provides PgAdmin.
- MySQL only supports standard data types (string, numeric, date, and time) while PostgreSQL supports advanced data types such as arrays, hstore, and user-defined data types.

Contributions:

Divyanshu Sharma : Front End

Navya .C : Schema and constraint change statement