Raptor Truck Rental

By: Database Warriors

Team:

Anusha Sama

Chandra Bhumireddy

Snehal Talole

Tushara Vepakomma

Overview:
We implemented a part of Truxxit website as our use case and designed a webpage called "Raptor Truck Rental." We used MY SQL as our data base and implemented the design using PHP. We connected MY SQL and PHP through XAMPP software. The webpage runs on the local host.

Use case:

- User will register using the sign-up link on the home page.
- User will be redirected to login page, where the user enters his/her credentials and system validates them.
- User can now check his/her current reservations or book a truck.
- When user clicks book truck, all the available trucks in the data base are displayed. User can select the truck and enters the start date and dropoff date.
- User logs back in and checks the current reservations.
- User can select pick up or cancel.
- After completing the reservation, invoice is generated based on actual start time and drop off time.
- Payment options are displayed based on the type of user. (regular or corporate)
- User completes the payment and is redirected to the ratings page.
- User gives the rating and is redirected to the home page.

Business Rules:

- Trucks are already present in DB.
- Every customer should have valid license.
- Customers can be either regular customers or corporate customers.
- If the customer type is corporate, then the company id and company name fields will be displayed on sign up page.
- The data base validates the user name and password fields when the user tries to log in.
- Trucks are displayed to the user along with the dimensions, base rate and model name. They are displayed based on the availability. Already booked truck is not displayed to the customer.
- Customer should be able to book one to many trucks and each truck can have one to many customers.
- Under current reservations, user will have option to cancel the reservation only if the select date is greater than 24 hours from the current date. The cancel option is disabled if the selected date is less than 24 hours.

- Pick up option is available until the drop off date given by the customer during booking the truck, so that user can pick up the truck whenever he/she wants.
- Once picked up, only complete reservation option is available for the users. User can complete the reservation whenever he/she wants.
- The invoice is generated after the reservation is completed. The total amount is calculated based on actual start date and actual drop off date.
- If the user is regular user, he/she can pay through credit card. If the user is a corporate user, the invoice is sent to the company which he/she mentions, and user will be displayed the message stating that the payment will be processed to the corporate account.
- The ratings submitted by the user at the end, will be save din data base with the respective invoice number.
- Base rate mentioned in the website in hourly based.

Triggers used:

The following triggers are written as part of the implementation of the use case. These triggers will perform the required CRUD operations on data base.

1. Trigger for updating the availability of the truck as soon as the user reserves the truck.

```
/* delimiter $$

drop trigger if exists update_availability $$

create trigger update_availability after insert on reservation

for each row begin

update truck

set Available = 0 where truck_id= new.Truck_id;

end$$

delimiter ; */
```

2. Trigger to update the availability to 1 if user cancels the reservation.

```
/* delimiter $$

drop trigger if exists update_truck $$

create trigger update_truck after update on reservation
```

```
for each row begin
if (new.cancel_status = 1 or new.actual_dropoffdate is not null) then
update truck
set Available = 1 where truck_id= new.Truck_id;
end if;
end$$
delimiter ; */
```

3. Triggers to insert the customers according to the type in either regular or corporate tables.

```
/*delimiter $$
drop trigger if exists insert_regular $$
create trigger insert_regular after insert on customer
for each row begin
if (new.Customer_type = "Regular") then
set @MaxRegID = (select max(reg_id) from regular);
insert into regular values(@MaxRegID+1, new.Customer_id,now());
end if;
end$$
delimiter ;
delimiter $$
drop trigger if exists insert_corporate $$
create trigger insert_corporate after insert on customer
for each row begin
if (new.Customer_type = "corporate") then
set @MaxcorpID = (select max(Corpo_id) from corporate);
insert into corporate (Corpo_id, customer_id, DOJ) values (@MaxcorpID+1, new.Customer_id,
now());
```

```
end if;
end$$
delimiter ; */
```

Stored Procedure used:

When user completes the reservation, the actual dropoff date is updated and invoice is generated. This is done by calling the stored procedure in PHP.

```
/*drop procedure insert_invoice;

delimiter //

create procedure insert_invoice

(IN id int)

begin

Update truckrental.reservation set actual_dropoffdate = now() where reservation_id = id;


set @randomId = concat(

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(id)*4294967296))*36+1, 1),

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),

	substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),
```

```
        substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),
        substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',
rand(@seed:=round(rand(@seed)*4294967296))*36+1, 1),
        substring('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789', rand(@seed)*36+1, 1)
       );

insert into invoice (invoice_no, reservation_id, payment_status ) values (@randomId, id, 0);
end //
delimiter ; */
```

Views:

The required attributes such as calculation of total price based on the base rate and actual select and drop dates are selected and a view is created for easy retrieval.

```
/*drop view if exists customer_invoice;
create view customer_invoice as
select r.Reservation_id as Reservation_No, t.model as Truckmodel, r.actual_selectdate as
startdate, r.actual_dropoffdate as enddate, t.baserate as baseprice,
                timestampdiff(hour, r.actual_selectdate,r.actual_dropoffdate) as
No_of_hours,((timestampdiff(minute, r.actual_selectdate,r.actual_dropoffdate))*BaseRate)/60
                as Total,concat(c.first_name,' ',c.last_name) as Name, c.email as email,
c.customer_type as type from truck t join reservation r on r.Truck_id = t.truck_id
                join customer c on c.customer_id = r.customer_id
*/
```

Demonstration:

The complete flow of the use case and all the validations are explained in detail in the video. Please follow the below link for more details.

https://www.youtube.com/watch?v=TYyiJfm_UF0&feature=youtu.be

Note: The actual video file along with MY SQL dump file and php code is uploaded to canvas.