CSE'22

SHIV NADAR UNIVERSITY

SNU SHUTTLE MANAGEMENT SYSTEM

1) SAI SRIRAM KALAVAGUNTA- 1810110203
2) LAKSHMI NARAYANA RAO BOYAPATI- 1810110115
3) NITIN SURYA MOTURU- 1810110150
4) V S SAI MAHENDRA KANIKICHERLA- 1810110268

ABSTRACT

As we all are aware of the SHUTTLE SERVICE to and fro the SNU campus, through this project we aim to simplify the ticket booking system for a hassle free booking and journey. Although an elegant idea, the present way of its implementation can be improved so as to simplify its management and also its usage by the student community at SNU.

Through this project we intend to provide a full-scale Website through which the SNU community can check the availability, price and successfully book the tickets for the shuttle. The website also aims to simplify the work for the management by listing clearly the details of the passengers and also arranging for a paper-free ticket collection.

This project has been an exemplary hands-on experience for all of us. Through this project we could apply in real life all that we had learnt in this course, we now have the confidence to go ahead and learn more advanced concepts in DBMS and apply them in real-life applications.

Content:

Introduction

In our project named SNU Shuttle Management System (SMS), we have created a dynamic website along with a database. This website has provisions for users of the Shuttle service and also for the transport committee; the users can book tickets, view their bookings, find details about routes and check availability and price. The administrators (admins) can view and approve/reject bookings for trips, add trips and routes view feedback given by the user and also do a lot more.

The main objective of this project is to ease the mode of transportation provided by SNU. As we know the booking system of the shuttle service in SNU is made in person. One of the members of the transport committee stays in the D-block to sell the tickets. The main aim of this project is to provide an online booking system. One of the main advantages of this project is that people can book tickets 24/7 so that they don't need to wait until the next day. They have the convenience to book whenever they want. We believe that this will also result in increase in the number of bookings. This will also save time for the transport committee. It increases the efficiency of operations as people don't have to wait in lines to book tickets. As this keeps track of all the reservations made it gives a better insight on how to expand the shuttle service, and with the feedback mechanism built into the website it should be easier for the committee to understand and resolve any problems.

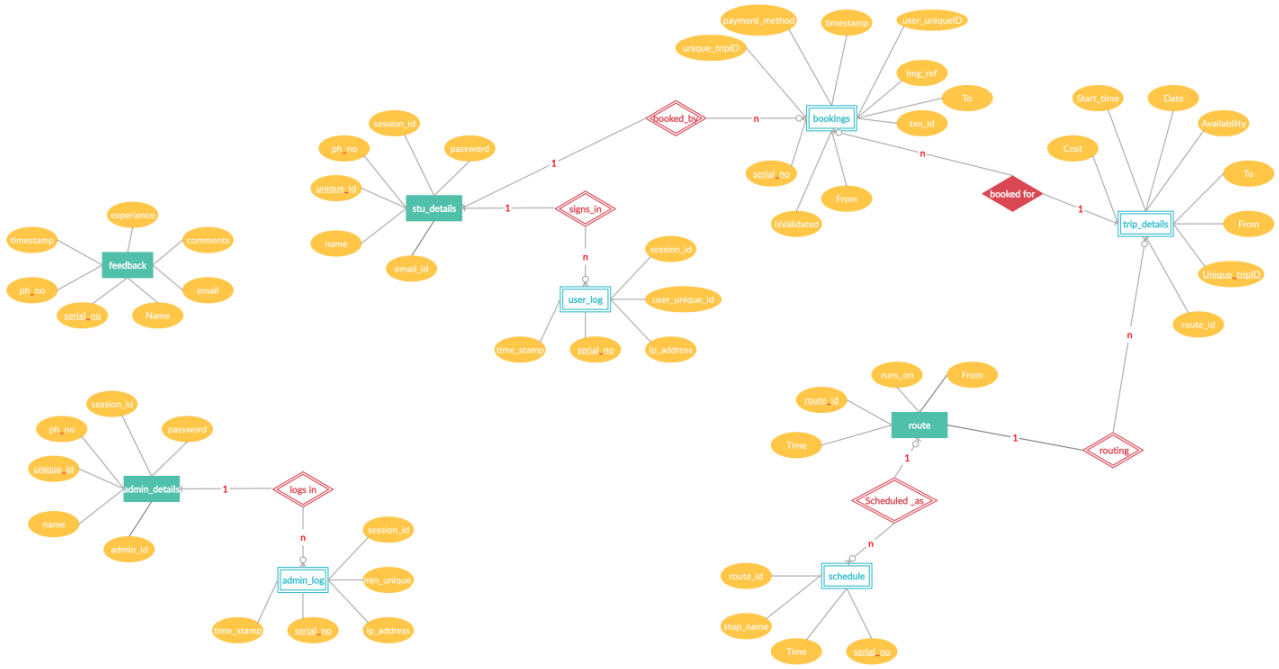
Main Objectives of this project are:

1. To provide a safe and easy way to book tickets for the shuttle service.
2. To implement the concepts taught in class in real world applications.
3. To have a hands on learning, while doing the project.
4. To show information regarding the shuttle service.
5. Convert the over-counter ticket sale system to an online system.
6. This will include an implementation of an online verification system for the payment.
7. Provision for paper-less Ticket Collection.
8. Provision for management to see who is boarding when and where.
9. Check seat availability before booking.
10. To build an interactive yet simple website.
11. To provide an online rating system for the SMS.
12. To manage your view bookings and also generate tickets for the bookings.
13. To take feedbacks from the customers.
14. This is made as user friendly as possible so that anyone with a little knowledge can use it
15. This project will reduce the tedious job of maintaining the paperwork by keeping all the project details of bus ticket booking tickets stored in the database, and giving real-time update on bookings to the admins.
16. To provide up-to-date information and other queries.
17. The objective of the project is to make easy the ticket booking system of the shuttle service simple, reliable, user friendly, and corrective. Moreover, less time consuming as compared to manual work.

Design: ER model and Relational Model

Entity-Relationship MODEL:

- Entity-Relationship Model or simply ER Model is a high-level data model diagram.
- In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand.
- It is also very easy for the developers to understand the system by just looking at the ER diagram.
- We use the ER diagram as a visual tool to represent an ER Model.



(please note that a copy of this image has been uploaded with this document)

NOTE:

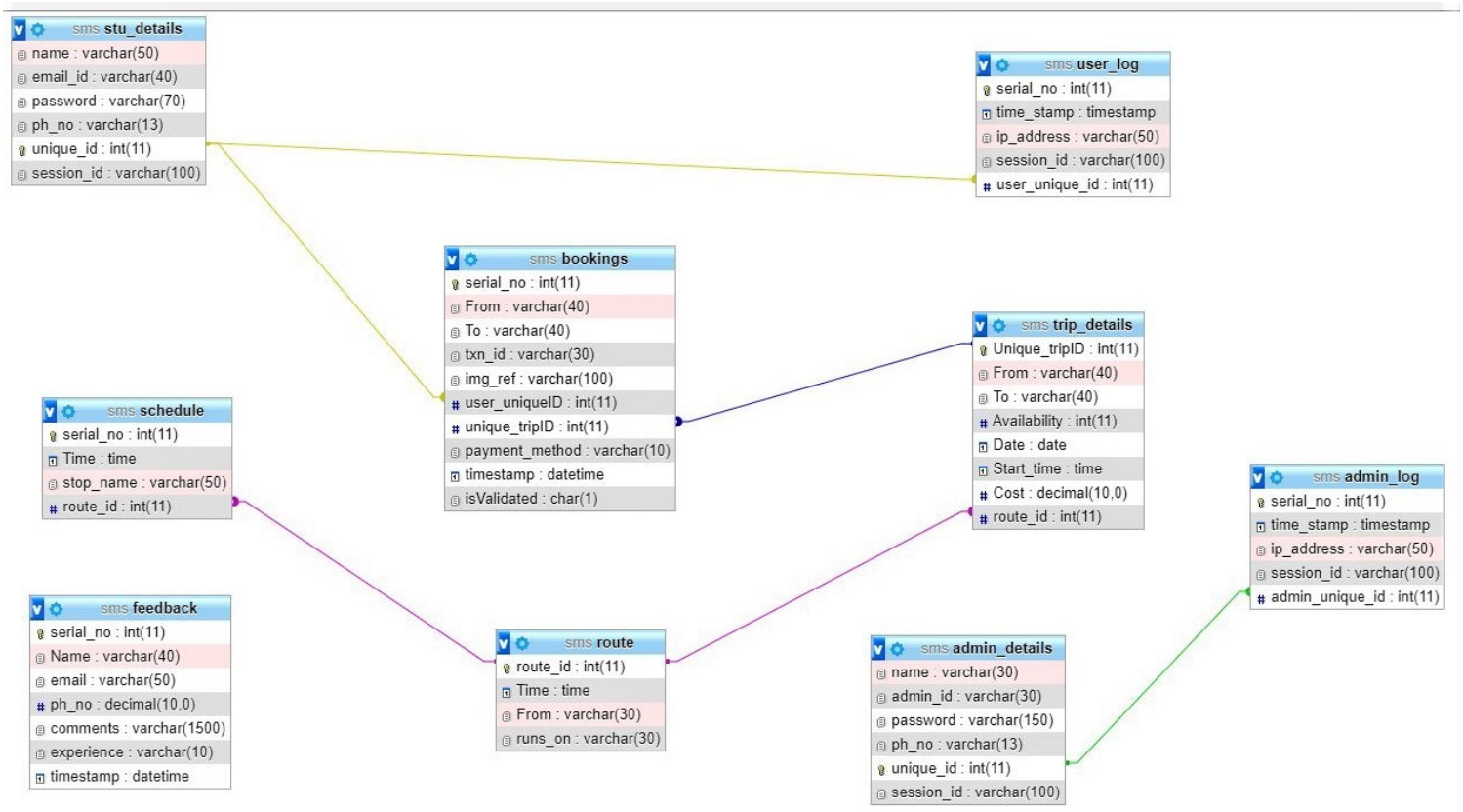
Cardinality is represented by 1:n format.

Total participation is represented by “o” at the end of the line joining relationship and entity. If there is no “o” that means it has partial participation.

Rest of the notations are as per the basic ER diagram notations.

RELATIONAL MODEL:

- Relational Model is the most widely used model.
- In this model, the data is maintained in the form of a two dimensional table.
- All the information is stored in the form of row and columns.
- The basic structure of a relational model is tables.



The lines joining the tables show that primary key of one table is introduced as foreign key in other table.

The backend used for this project is mysql command client and workbench.

Website and database linking was done with php and node js.

Database used is “SMS” meaning shuttle management system.

The tables created under this database are:

1. admin_details
2. admin_log
3. bookings
4. feedback
5. route
6. schedule
7. stu_details
8. trip_details
9. user_log

A total of 9 tables were used.

Different kinds of data types are used namely int, time, varchar, datetime, timestamp, decimal and char.

Front-End and Back-End Implementation

In this project we have developed an interactive website, which we believe takes on our objective to make the booking, ticket collection and its verifications, a smooth, hassle free and more importantly a online process. The Shuttle Management System (SMS), as we call it consists of two separate websites- one for users and the other for admins. Each of the website has different pages and UI appropriately designed.

Coming to the implementation of the website, our website like most other websites has three parts- the front-end, the server and the back-end.

The front-end mostly consists of HTML5, JavaScript and most of the styling done using CSS3.The Backend is done in MySQL (technically MariaDB, 100% compatible with MySQL). This website presently runs on an apache server coming directly out of XAMPP, a cross-platform web server solution. Any and all access to the database through the website is done using PHP.

FRONT END:

The part in which the user can interact easily in a website, is commonly known as front-end of a website. Everything we see when we moving on internet from fonts, colours, menus, links etc. are the combo of front-end languages controlled by our computer browsers. To develop these there are many platforms like WordPress, notepad, bootstrap etc. and also there are many text editors like notepad, WebStorm, Visual Studio Code, etc. The need for any of these depends on which our problem basically depends on. Users should be able to understand which tools among the above are best fit for specific marks depending on their usage. Coming to our project we mainly used three languages that is HTML5, CSS3 and JavaScript, to design our front-end website.

Html which is also known as Hyper Text Markup Language is the back bone of any website. Without this webpage does not exist. Hypertext generally means that text has links to navigate in it. When anyone clicks on it, it will redirect to another page or site. And the mark-up language usually indicates like the text can be made into tables, images, buttons, etc. It is the HTML code that provides an overall framework of how the site will look.

Then coming to CSS, the abbreviation for Cascading Style Sheets. CSS controls the presentation aspect of any website. It is responsible for the styles that we are seeing when we open a webpage. It allows our website to have its own unique look. It is responsible for maintaining style sheets and are triggered based on the inputs like size of the screen, resolution, adjusting font size and font colour, etc.

Although we used some JavaScript throughout our project but JavaScript is an event-based imperative programming language. It is used to transform a static html page into a dynamic interface. It is actually used to manipulate a web page in response to events like when we click on a button to travel along somewhere, to get the inputs given by users, etc.

We used a common text editor that is notepad for dealing with above languages. We believed using a simple text editor is a good way to learn HTML, CSS and JavaScript. Coding in it in fact made it easier to achieve our goals. The entire front end of our project has been coded in the notepad.

These are the entire details of our front end implementation. The following are some of the snippets in our website done in notepad.

*initial.html - Notepad

File Edit Format View Help

```
<html>
<body>
  <div class="container">
  <span class="text1">Welcome to</span>
  <span class="text2">SNU Shuttle Mangement Service</span>
</div>

</body>

</html>
```

*feedback.html - Notepad

File Edit Format View Help

```
<!DOCTYPE html>
<body>
</body>
<script>
  function myFunction() {
  alert("Please Go To Home Page To See Our Details!!");
}
</script>

</html>
```

*thankyou.html - Notepad

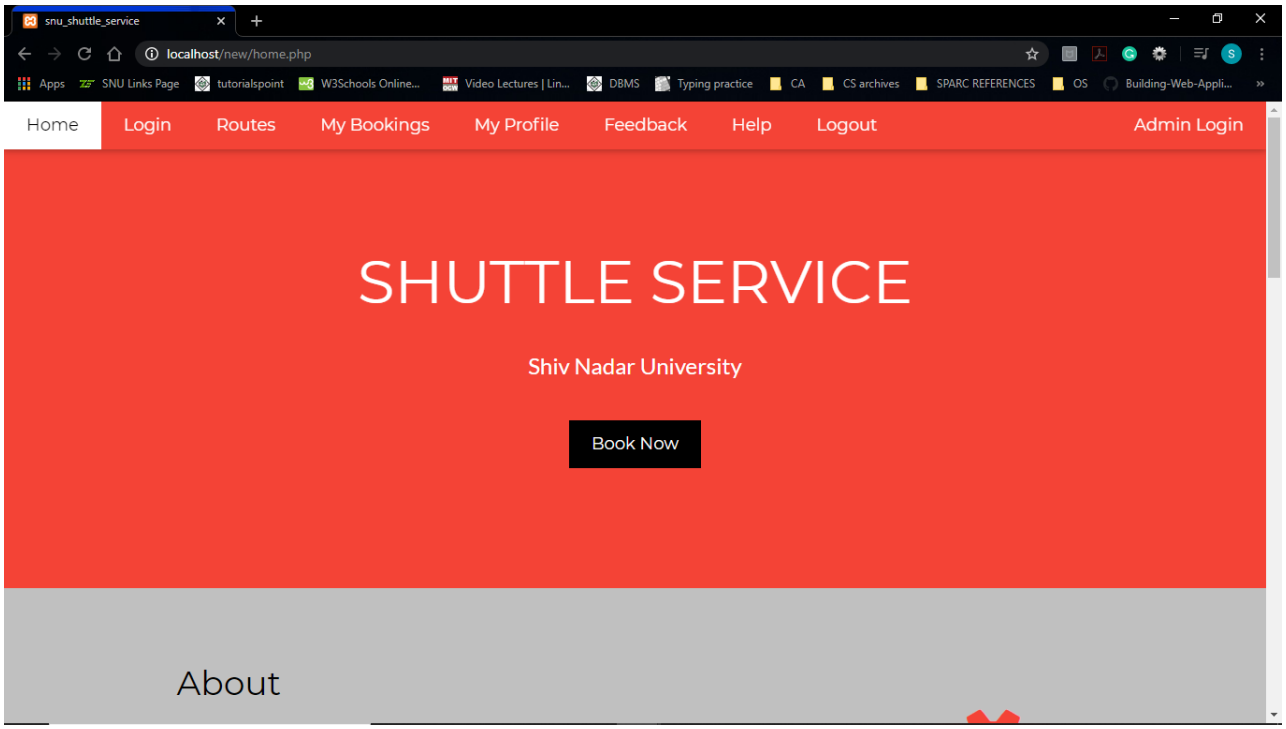
File Edit Format View Help

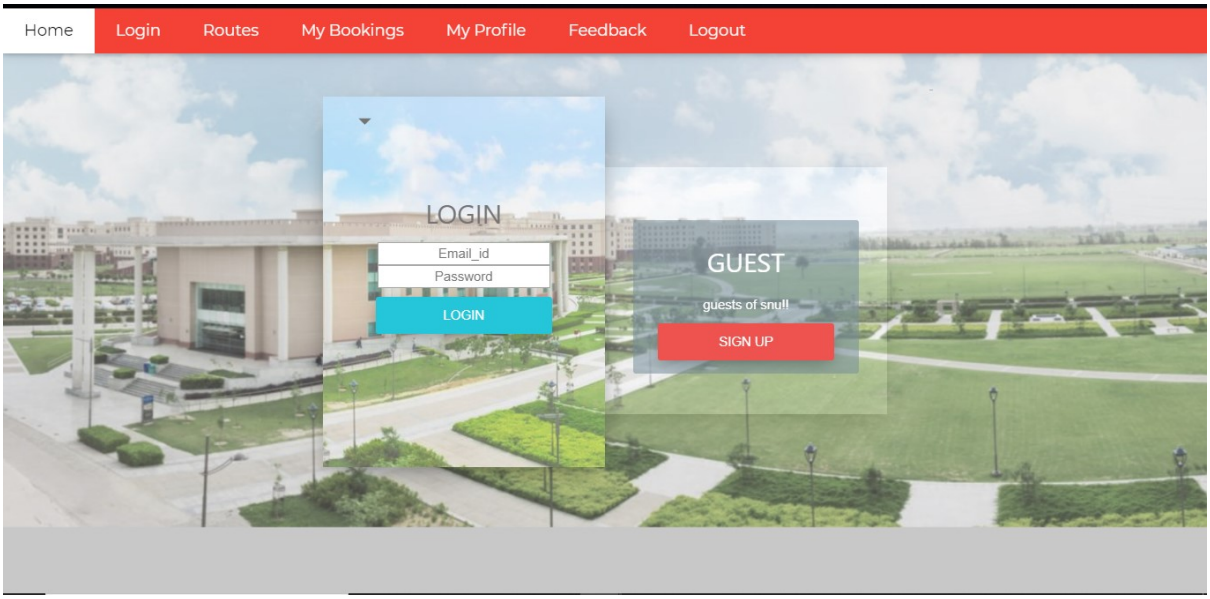
```
<html>
   <head>
      <title>Thank_You</title>
     </head>
<style>
.w3-bar {font-family: "Montserrat", sans-serif}
.image{
background-image: url("https://pngimage.net/wp-content/uploads/2018/06/nice-background-png-1.png");
            background-repeat: no-repeat;
            background-position: center center;
            background-size: cover;
            background-attachment: fixed;
            position: fixed;
            top: 0;
            left: 0;
            right: 0;
            bottom: 0;
}
.text{
  position: relative;
 text-align: center;
  top: 200px;
font-size: 55px;
}
```

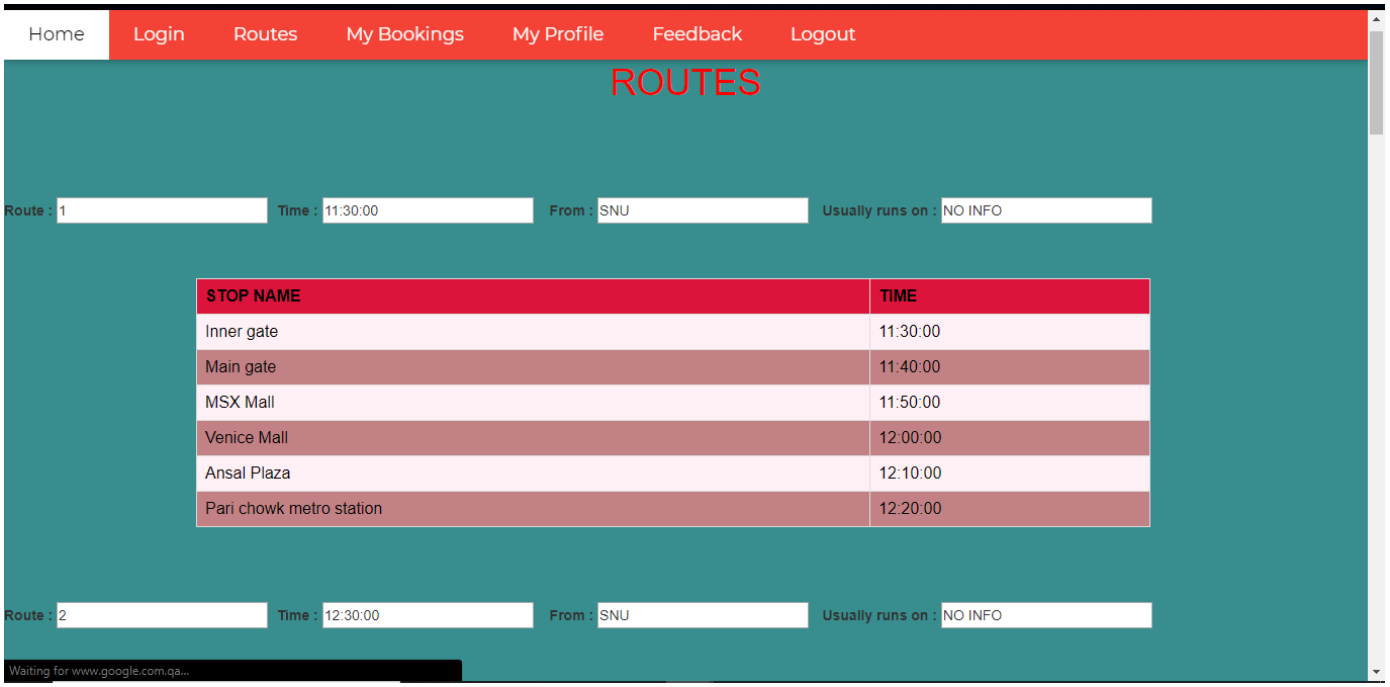
We have referred to the website w3schools in case of any doubts while coding our website. This made it easy for us to complete the project. Therefore, we successfully implemented the front end using these resources.

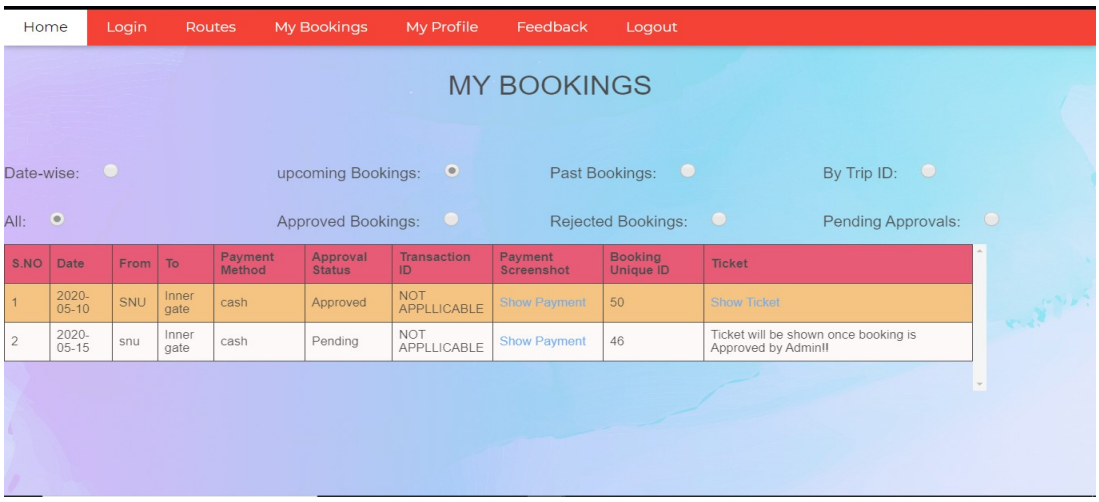
Here we have some screenshots of few webpages from our website:

User site:

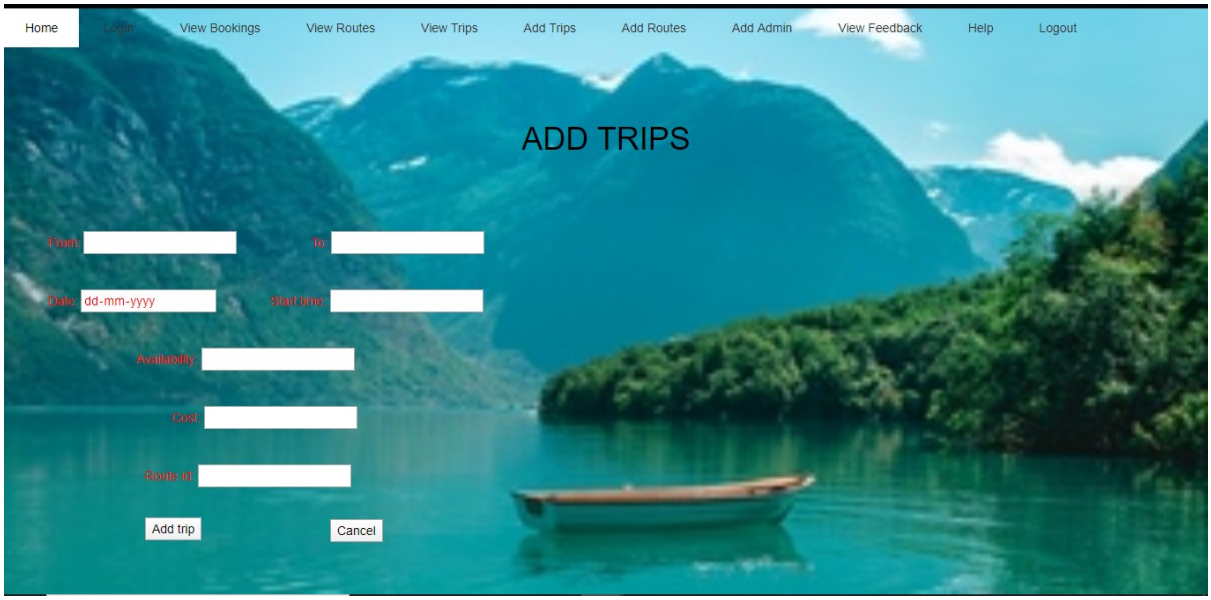


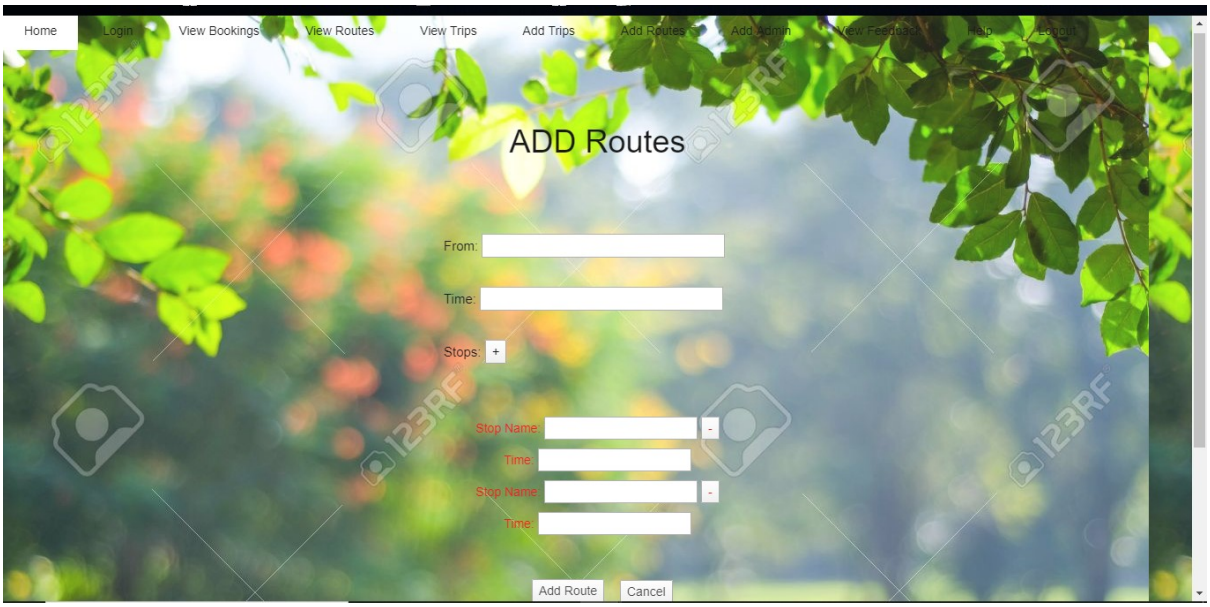

Home
Login
Routes
My Bookings
My Profile
Feedback
Logout
LOGIN
Email_id
Password
LOGIN
GUEST
guests of snu!!
SIGN UP


Home
Login
Routes
My Bookings
My Profile
Feedback
Logout
ROUTES
Route : 1
Time : 11:30:00
From : SNU
Usually runs on : NO INFO
STOP NAME
TIME
Inner gate
11:30:00
Main gate
11:40:00
MSX Mall
11:50:00
Venice Mall
12:00:00
Ansal Plaza
12:10:00
Pari chowk metro station
12:20:00
Route : 2
Time : 12:30:00
From : SNU
Usually runs on : NO INFO
Waiting for www.google.com.qa...


Home
Login
Routes
My Bookings
My Profile
Feedback
Logout
MY BOOKINGS
Date-wise:
upcoming Bookings:
Past Bookings:
By Trip ID:
All:
Approved Bookings:
Rejected Bookings:
Pending Approvals:
S.NO
Date
From
To
Payment Method
Approval Status
Transaction ID
Payment Screenshot
Booking Unique ID
Ticket
1
2020-05-10
SNU
Inner gate
cash
Approved
NOT APPLLICABLE
Show Payment
50
Show Ticket
2
2020-05-15
snu
Inner gate
cash
Pending
NOT APPLLICABLE
Show Payment
46
Ticket will be shown once booking is Approved by Admin!!

From admin site:





BACK-END:

Now coming to the Backend- The Database, as already mentioned the database has been constructed in MySQL, and we believe it matches the requirements of the Shuttle Management System. The database as seen in the ER diagram and the relational model consists of 9 tables.

We have followed the standard procedure taught to us in the course, step by step designing our database and then the website.

Firstly we talked to members of the transport committee, who explained us the due process of the current ticket booking system and also listed to us all their

requirements from a online model. Then we started discussing about how our website should look like, after some successful discussions we then designed an ER model for our database. After the ER model was ready we then described each entity into a table, and hence slowly arrived at the relational model.

Looking at the raw design of database, we applied the concepts taught to us in the class to reduce redundancy in the database and also make it as close as possible to the 3NF form.

Then we created the database in MySQL and step by step created tables, with suitable primary and foreign keys. After making the database we checked if all our requirements database-wise were being met. After we were satisfied we then started with designing the website, and connecting the front-end and back-end through the apache server.

Both the database and the website are protected against SQL Injection by preparing (in PHP) the statements before execution, and against HTML Injection where ever required.

The passwords are salted MD5 encrypted and stored in their hashed forms in the database. This website also tries to maintain concurrency by using transactions where ever appropriately required. Also we would like to add that the website prevents simultaneous multiple logins by the same user, in order to protect the database from storing repetitive, redundant or incorrect data.

The database also has a user log and a admin log which notes the login activity of the users and admins.

Also note that a user needs to be signed in only to book tickets or view personal information, but a admin needs to log in to pretty much do anything in the admin's website.

Data Dictionary of all tables in database:

1. **admin_details**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
name	varchar(30)	Yes	*NULL*			
admin_id	varchar(30)	Yes	*NULL*			
password	varchar(150)	Yes	*NULL*			
ph_no	varchar(13)	Yes	*NULL*			
unique_id *(Primary)*	int(11)	No				
session_id	varchar(100)	Yes	0			

2. **admin_log**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
serial_no *(Primary)*	int(11)	No				
time_stamp	timestamp	No	current_timestamp()			
ip_address	varchar(50)	Yes	*NULL*			
session_id	varchar(100)	Yes	*NULL*			
admin_unique_id	int(11)	Yes	*NULL*	admin_details -> unique_id		

3. **bookings**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
serial_no *(Primary)*	int(11)	No				
From	varchar(40)	Yes	*NULL*			
To	varchar(40)	Yes	*NULL*			
txn_id	varchar(30)	Yes	*NULL*			
img_ref	varchar(100)	Yes	*NULL*			
user_uniqueID	int(11)	Yes	*NULL*	stu_details -> unique_id		
unique_tripID	int(11)	Yes	*NULL*	trip_details -> Unique_tripID		
payment_method	varchar(10)	Yes	*NULL*			
timestamp	datetime	Yes	*NULL*			
isValidated	char(1)	No	P			

4. **feedback**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
serial_no *(Primary)*	int(11)	No				
Name	varchar(40)	Yes	*NULL*			
email	varchar(50)	Yes	*NULL*			
ph_no	decimal(10,0)	Yes	*NULL*			
comments	varchar(1500)	Yes	*NULL*			
experience	varchar(10)	Yes	*NULL*			
timestamp	datetime	Yes	*NULL*			

5. **route**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
route_id *(Primary)*	int(11)	No				
Time	time	Yes	*NULL*			
From	varchar(30)	Yes	*NULL*			
runs_on	varchar(30)	Yes	NO INFO			

6. **schedule**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
serial_no *(Primary)*	int(11)	No				
Time	time	Yes	*NULL*			
stop_name	varchar(50)	Yes	*NULL*			
route_id	int(11)	Yes	*NULL*	route -> route_id		

7. **stu_details**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
name	varchar(50)	No				
email_id	varchar(40)	Yes	*NULL*			
password	varchar(70)	No				
ph_no	varchar(13)	No				
unique_id *(Primary)*	int(11)	No				
session_id	varchar(100)	Yes	0			

8. **trip_details**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
Unique_tripID *(Primary)*	int(11)	No				
From	varchar(40)	Yes	*NULL*			
To	varchar(40)	Yes	*NULL*			
Availability	int(11)	Yes	*NULL*			
Date	date	Yes	*NULL*			
Start_time	time	Yes	*NULL*			
Cost	decimal(10,0)	Yes	*NULL*			
route_id	int(11)	Yes	*NULL*	route -> route_id		

9. **user_log**

Column	Type	Null	Default	Links to	Comments	Media (MIME) type
serial_no *(Primary)*	int(11)	No				
time_stamp	timestamp	Yes	*NULL*			
ip_address	varchar(50)	Yes	*NULL*			
session_id	varchar(100)	Yes	*NULL*			
user_unique_id	int(11)	Yes	*NULL*	stu_details -> unique_id		

The following are the screenshots of some tables in our database with live data:

1. Stu_details: shows student details (used for login)

name	email_id	password	ph_no	unique_id	session_id
Sai Krishna Karthik	dk984@snu.edu.in	ed7609b36a1d83f3eec1879ad0b880f4	9121696999	1001	0
Ben Tennison	bt010@snu.edu.in	724ba3015486ab7b33c826029b88a7aa	9876543210	1002	0
Peter Parker	pp456@snu.edu.in	29fbb59d1896848ee35069469a8cfda3	7896547320	1003	0
Romannof	bw345@snu.edu.in	0e7ac88136b05e8694db19c1fad0d4c8	9867543260	1004	0
Ajay Kumar	ak047@snu.edu.in	675865b4cf4c8a5dcefc801f38695041	8679540321	1005	0
t	t@e.c	c7466d26115fabcab51ed3cd6e1ce82f	9963829696	1006	vh4ir0ccblmo2gj50dos6tk97j
mahendra	vk767@snu.edu.in	176b5fb9a2677d6dc952dbacaf6a5bcd	9247541234	1007	0s0smnjtrf50tqls86t4leq9j2

Please note that the password has been stored in its hashed (Salted MD5) format.

2. Schedule: depicting the whole route related to route_id

serial_no	Time	stop_name	route_id
1	11:30:00	Inner gate	1
2	11:40:00	Main gate	1
3	11:50:00	MSX Mall	1
4	12:00:00	Venice Mall	1
5	12:10:00	Ansal Plaza	1
6	12:20:00	Pari chowk metro station	1
7	12:30:00	airport	2
8	12:40:00	terminal 3	2

3. Bookings : records all bookings made

serial_no	From	To	txn_id	img_ref	user_uniqueID	unique_tripID	payment_method	timestamp	isValidated
24	depot station	SNU	0	NULL	1008	5	cash	2020-04-27 14:18:13	P
25			0	NULL	1006	1	cash	2020-04-27 14:22:41	A
26	SNU	airport	0	NULL	1006	1	cash	2020-04-28 21:58:31	P
27	depot station	SNU	0	NULL	1008	5	cash	2020-04-29 15:51:12	P
28			0	NULL	1001	5	cash	2020-04-29 21:09:10	P
29	depot station	SNU	22341323212133	Sai Krishna Karthik_5_2020-04-29 211016.jpg	1001	5	paytm	2020-04-29 21:10:17	P
30	SNU	sector 16	0	NULL	1001	2	cash	2020-04-29 21:25:22	A
31	SNU	botanical garden	0	NULL	1008	2	cash	2020-04-30 18:48:09	P
32	snu	Ansal Plaza	123456781234	ssr_13_2020-05-01 175921.jpg	1008	13	paytm	2020-05-01 17:59:21	A

Please note that the img_ref and txn_id are set to null in case user opts for cash payment method.

Moving on to the server, as already mentioned it runs on a apache distribution provided by XAMPP. We have used PHP as the server side scripting language.

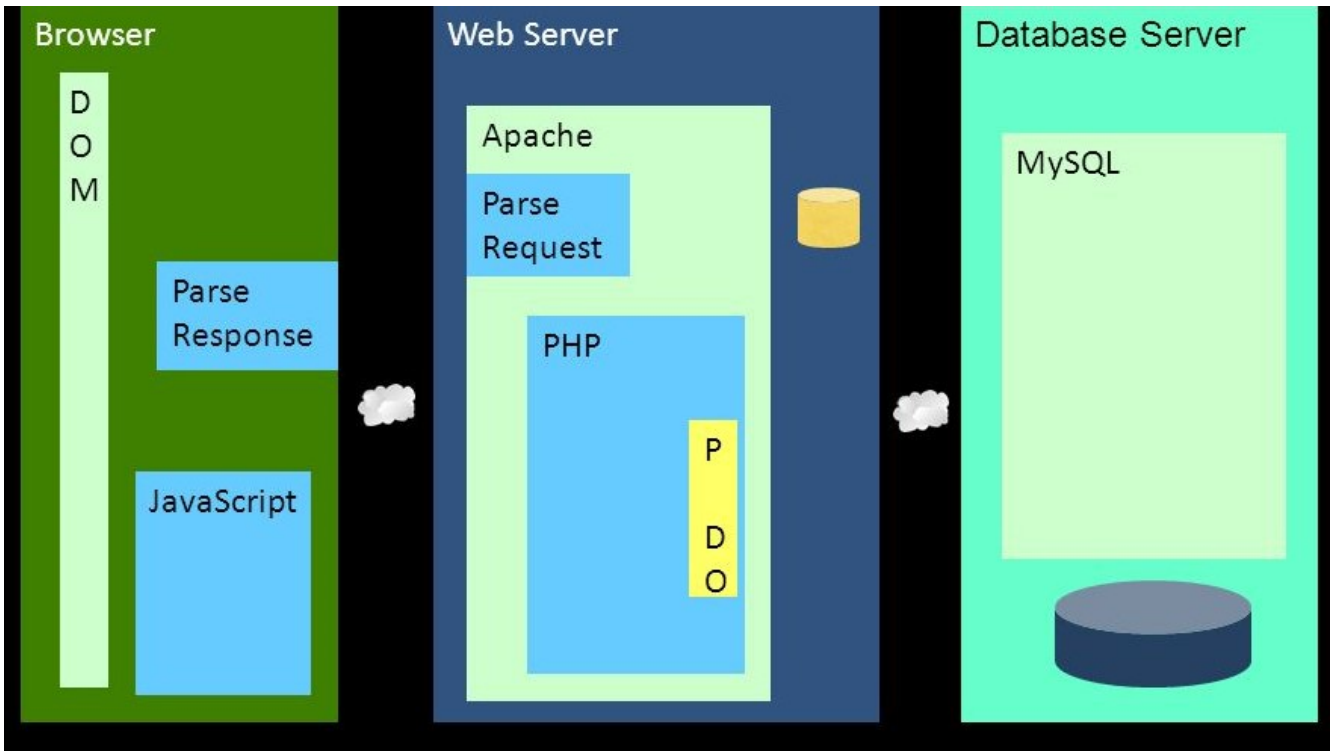
```php
<?php
    require_once "pdo.php";
    session_start();

    $id=session_id();
    if(isset($_SESSION['name']) && isset($_SESSION['admin_id']) && isset($_SESSION['adminID']))
    {  $sql = "SELECT session_id FROM admin_details WHERE admin_id=:id";
        //echo($_SESSION['email_id']);
        $stmt = $pdo->prepare($sql);
        $stmt->execute(array(':id' => $_SESSION['admin_id']));
        $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if($row[0]['session_id']==$id)
        {
            //var_dump($row);
        }
        else
        {
      echo("Multiple Logins From Same Admin Detected...Latest can continue. You are being LOGGED-OUT.");
      sleep(2);
      session_unset();
      header( 'Location: logout.php' ) ;
      return;
        }

    }
      else {echo("ACCESS DENIED!! LOGIN FIRST");
      sleep(2);
      header( 'Location: login.php' ) ;
      return;
    }
```

```php
require_once "pdo.php";
session_start();
//if ( !isset($_POST['datesel']) )
 //return;
//echo('present'.$_POST['date']);
if(isset($_POST['sel']))
{
$sql="SELECT * FROM ((`bookings`
INNER JOIN trip_details  ON bookings.unique_tripID=trip_details.Unique_tripID)
INNER JOIN stu_details ON bookings.user_uniqueID=stu_details.unique_id)  where isValidated=:isv and trip_details.Date >= CURDATE()
";//and trip_details.Start_time > CURTIME()
$stmt = $pdo->prepare($sql);
$stmt->execute(array(':isv'=>$_POST['sel']));
$rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
if($rows==false)
{echo"<p>No Bookings found.</p>";}
else{

    echo"<table border=\"1\" id=\"bookingsTable\">
```

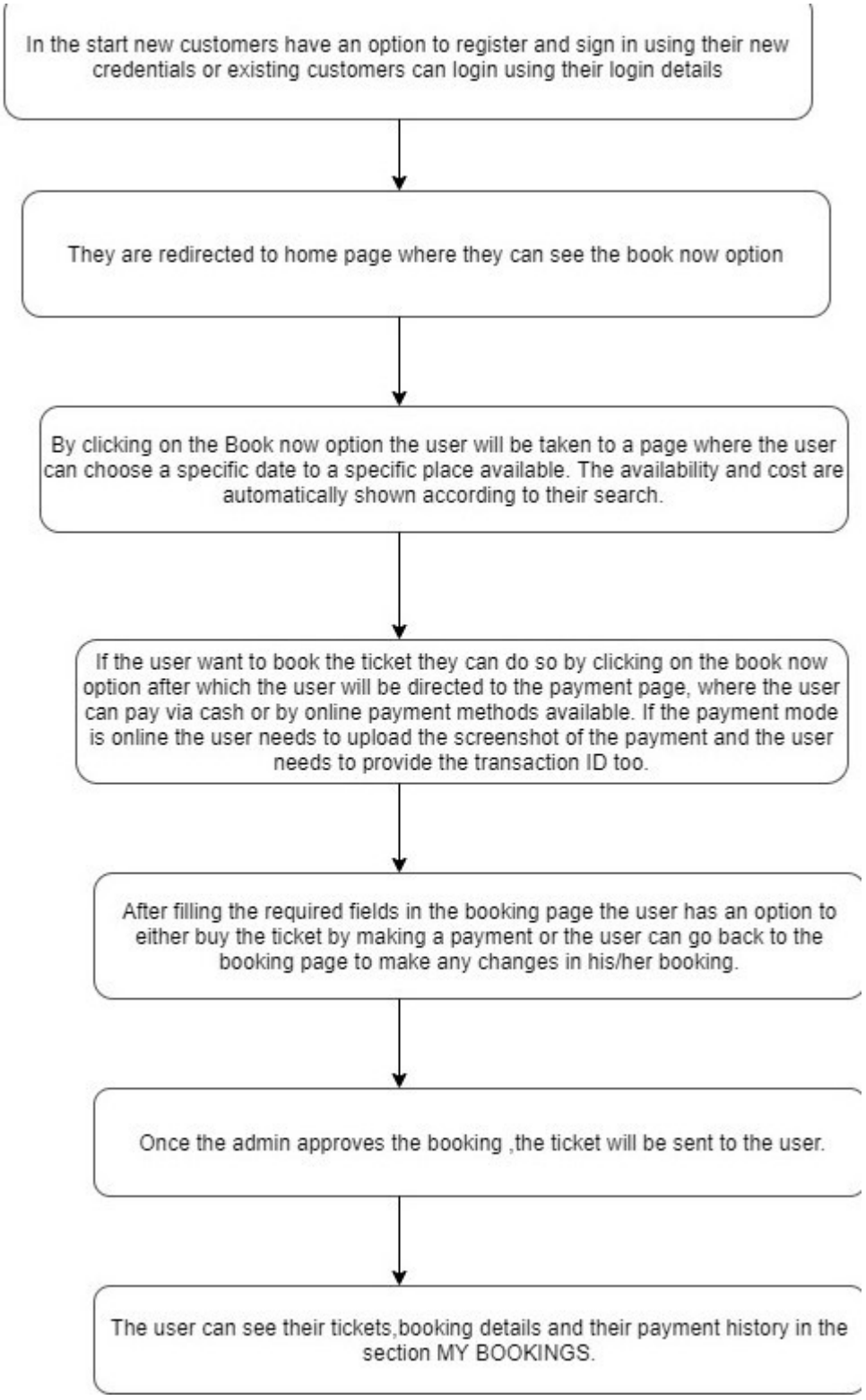
```php
$sql="start transaction;";
$stmt = $pdo->prepare($sql);
$stmt->execute();

$sql1 = "SELECT CURRENT_TIMESTAMP();";
        $stmt1 = $pdo->prepare($sql1);
            $stmt1->execute();
            $time = $stmt1->fetchAll(PDO::FETCH_ASSOC);

$sql="INSERT INTO bookings (`timestamp`,`From`,`To`,txn_id,img_ref,payment_method,user_uniqueID,unique_tripID) values
  (:ts,:selected_from,:selected_to,:txn_id,:img_ref,:paymeth,:user,:trip)";
$stmt = $pdo->prepare($sql);
$stmt->execute(array(':ts'=>$time[0]['CURRENT_TIMESTAMP()'],':selected_from' => $_GET['From'], ':selected_to' => $_GET['To'],':txn_id'
  =>$_SESSION['txn_id'],':img_ref'=>$_SESSION['fileLoc'],':paymeth'=>$_SESSION['paymeth'],':user'=>$_SESSION['userID'],':trip'=>$_
  SESSION['Unique_tripID']));


$sql = "UPDATE trip_details SET Availability=Availability-1 WHERE Unique_tripID=:trip";
$stmt = $pdo->prepare($sql);
$stmt->execute(array(':trip'=>$_SESSION['Unique_tripID']));

$sql="commit;";
$stmt = $pdo->prepare($sql);
$stmt->execute();

    //$_SESSION['success_msg']="Ticket Booked Successfully.";
  header( 'Location: thankyou.html' ) ;
  return;

?>
```

```php
$sql="SELECT DISTINCT stop_name FROM `schedule` INNER JOIN trip_details ON schedule.route_id=trip_details.route_id where trip_details.
    Date=:selected_date and trip_details.From=:selected_from";
$stmt = $pdo->prepare($sql);
$stmt->execute(array(':selected_date' => $selected_date, ':selected_from'=>$selected_from));
$row = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Our website Model is depicted as-



Flow-Chart showing the way through our website:



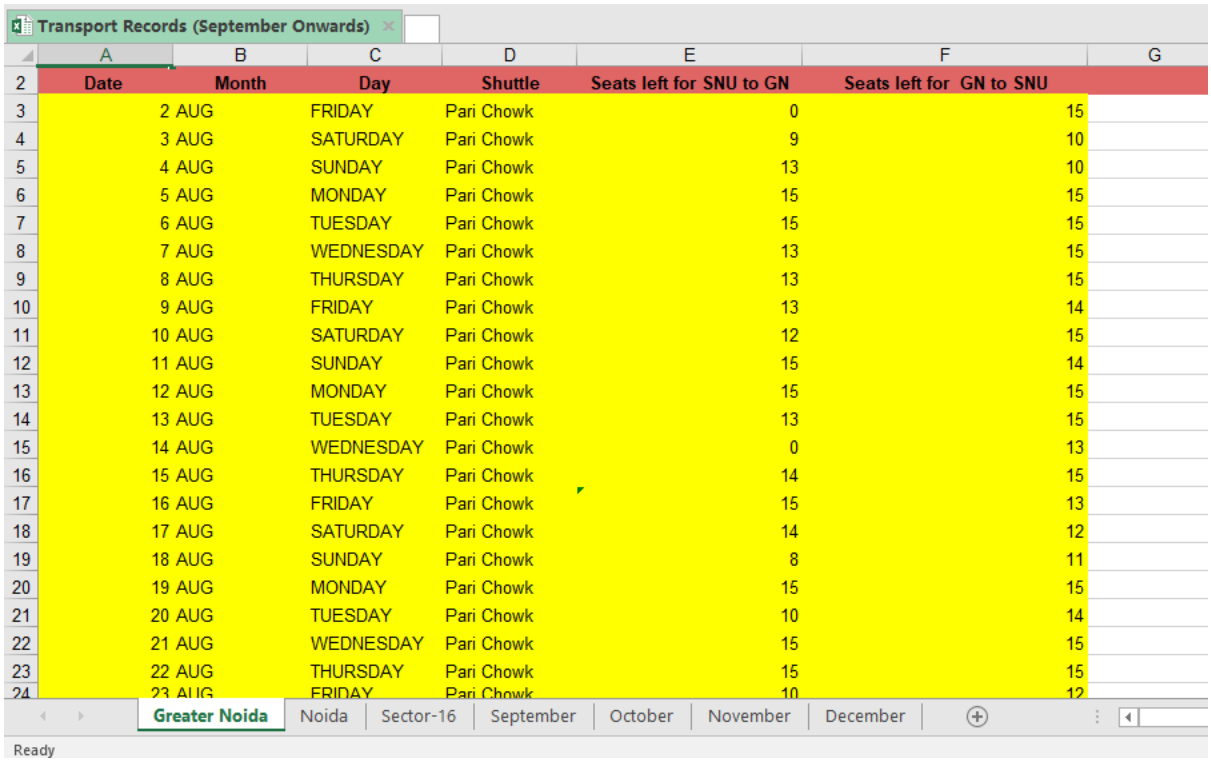
Experimental study, Results and Analysis

The project as already mentioned aims to simplify and take online the booking process of the Shuttle services to and fro campus. To make this possible we have been talking with some of the members of the SNU Transport committee. Their inputs have been considered at all stages and the website has been shaped in a way that it helps them out by simplifying their work.

The route plan for one of the shuttle service:

SHUTTLE SERVICE TO GRETAER NOIDA AND NOIDA TIMINGS		
(Except start time, all other timings may vary depending on traffic conditions)		
SHUTTLE : SNU - GREATER NOIDA & BACK (MONDAY TO SUNDAY)		
To Pari Chowk Metro Station	**Place**	**From Pari Chowk Metro Station**
Departure		**Departure**
3:30 PM	SNU FH	9:00 PM
3:35 PM	Inner Gate	8:55 PM
3:40 PM	Main Gate	8:50 PM
4:10 PM	MSX Mall	8:20 PM
4:15 PM	Venice Mall	8:15 PM
4:25 PM	Ansal Plaza	8:05 PM
4:30 PM	Pari Chowk Metro Station	8:00 PM
WEEKEND SHUTTLE : SNU - NOIDA & BACK (SATURDAY & SUNDAY)		
To Noida (Footover Bridge at GIP Mall)	**Place**	**From Noida (Footover Bridge at GIP Mall)**
Departure		**Departure**
10:00 AM	SNU FH	7:50 PM
10:05 AM	Inner Gate	7:45 PM
10:10 AM	Main Gate	7:40 PM
10:40 AM	Venice Mall	7:10 PM
10:50 AM	Ansal Plaza	7:00 PM
11:20 AM	GIP Mall	6:30 PM

We have received some excel sheets in which previous trips and their bookings were recorded.

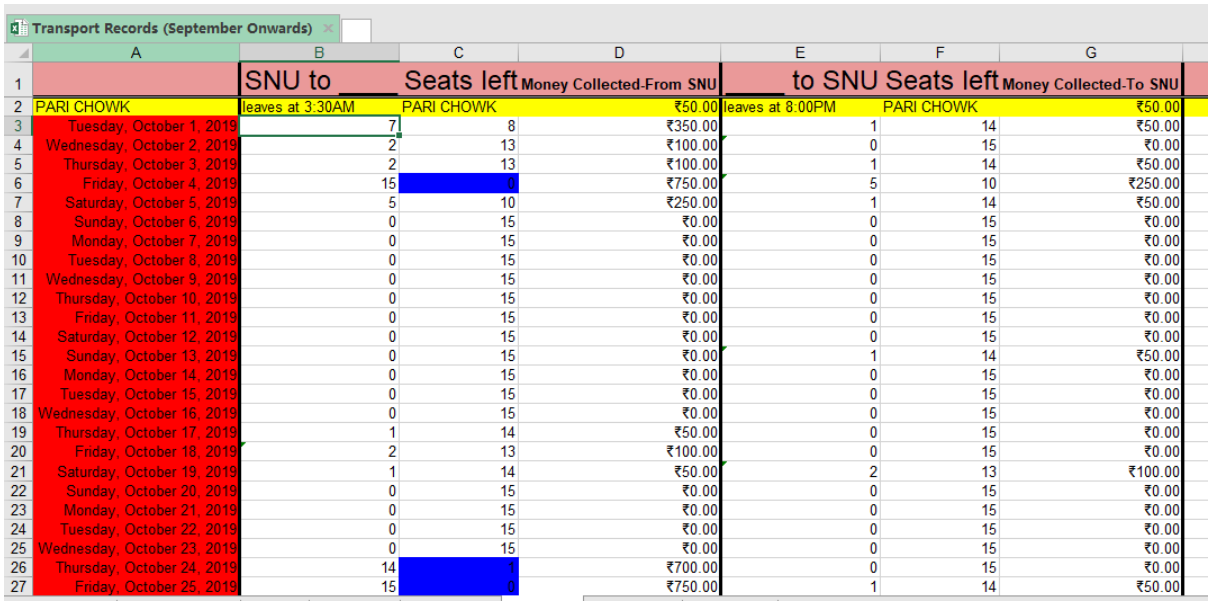


Transport Records (September Onwards)

	A	B	C	D	E	F	G
2	Date	Month	Day	Shuttle	Seats left for SNU to GN	Seats left for GN to SNU	
3	2	AUG	FRIDAY	Pari Chowk	0	15	
4	3	AUG	SATURDAY	Pari Chowk	9	10	
5	4	AUG	SUNDAY	Pari Chowk	13	10	
6	5	AUG	MONDAY	Pari Chowk	15	15	
7	6	AUG	TUESDAY	Pari Chowk	15	15	
8	7	AUG	WEDNESDAY	Pari Chowk	13	15	
9	8	AUG	THURSDAY	Pari Chowk	13	15	
10	9	AUG	FRIDAY	Pari Chowk	13	14	
11	10	AUG	SATURDAY	Pari Chowk	12	15	
12	11	AUG	SUNDAY	Pari Chowk	15	14	
13	12	AUG	MONDAY	Pari Chowk	15	15	
14	13	AUG	TUESDAY	Pari Chowk	13	15	
15	14	AUG	WEDNESDAY	Pari Chowk	0	13	
16	15	AUG	THURSDAY	Pari Chowk	14	15	
17	16	AUG	FRIDAY	Pari Chowk	15	13	
18	17	AUG	SATURDAY	Pari Chowk	14	12	
19	18	AUG	SUNDAY	Pari Chowk	8	11	
20	19	AUG	MONDAY	Pari Chowk	15	15	
21	20	AUG	TUESDAY	Pari Chowk	10	14	
22	21	AUG	WEDNESDAY	Pari Chowk	15	15	
23	22	AUG	THURSDAY	Pari Chowk	15	15	
24	23	AUG	FRIDAY	Pari Chowk	10	12	

Greater Noida | Noida | Sector-16 | September | October | November | December

Ready



Transport Records (September Onwards)

	A	B	C	D	E	F	G
1		SNU to ____	Seats left	Money Collected-From SNU	____ to SNU	Seats left	Money Collected-To SNU
2	PARI CHOWK	leaves at 3:30AM	PARI CHOWK	₹50.00	leaves at 8:00PM	PARI CHOWK	₹50.00
3	Tuesday, October 1, 2019	7	8	₹350.00	1	14	₹50.00
4	Wednesday, October 2, 2019	2	13	₹100.00	0	15	₹0.00
5	Thursday, October 3, 2019	2	13	₹100.00	1	14	₹50.00
6	Friday, October 4, 2019	15	0	₹750.00	5	10	₹250.00
7	Saturday, October 5, 2019	5	10	₹250.00	1	14	₹50.00
8	Sunday, October 6, 2019	0	15	₹0.00	0	15	₹0.00
9	Monday, October 7, 2019	0	15	₹0.00	0	15	₹0.00
10	Tuesday, October 8, 2019	0	15	₹0.00	0	15	₹0.00
11	Wednesday, October 9, 2019	0	15	₹0.00	0	15	₹0.00
12	Thursday, October 10, 2019	0	15	₹0.00	0	15	₹0.00
13	Friday, October 11, 2019	0	15	₹0.00	0	15	₹0.00
14	Saturday, October 12, 2019	0	15	₹0.00	0	15	₹0.00
15	Sunday, October 13, 2019	0	15	₹0.00	1	14	₹50.00
16	Monday, October 14, 2019	0	15	₹0.00	0	15	₹0.00
17	Tuesday, October 15, 2019	0	15	₹0.00	0	15	₹0.00
18	Wednesday, October 16, 2019	0	15	₹0.00	0	15	₹0.00
19	Thursday, October 17, 2019	1	14	₹50.00	0	15	₹0.00
20	Friday, October 18, 2019	2	13	₹100.00	0	15	₹0.00
21	Saturday, October 19, 2019	1	14	₹50.00	2	13	₹100.00
22	Sunday, October 20, 2019	0	15	₹0.00	0	15	₹0.00
23	Monday, October 21, 2019	0	15	₹0.00	0	15	₹0.00
24	Tuesday, October 22, 2019	0	15	₹0.00	0	15	₹0.00
25	Wednesday, October 23, 2019	0	15	₹0.00	0	15	₹0.00
26	Thursday, October 24, 2019	14	1	₹700.00	0	15	₹0.00
27	Friday, October 25, 2019	15	0	₹750.00	1	14	₹50.00

The bookings and the amount of money collected till now was recorded in these forms manually, which is difficult to maintain.

Our website automatically maintains the database and for back-up purposes we has also made available the provision to export the bookings and related info into excel sheets.

This website has been tested well by our group and also some friends.

Further Enhancements

Any work is never perfect in this ever evolving world, and our project is no exception. Although we have tried our best, we would like to list a few improvements which can be done to make the website more suitable and user-friendly.

1. Though the encryption used is salted MD5, it is considered weak as per industry standards. It is sufficient for the model website but SHA-256 or other suitable encryption methods should be deployed for better protection.

2. The website doesn't give the option to users for modifying their details such as password, or phone number ,etc. This can be provided.

3. Although the website manages privacy at a basic level, the database and server should be inaccessible by the users or even by admins to ensure this.

4. QR codes or barcodes can be added for UPI payments.

5. An automatically generated mail can be sent from server whenever the payment has been approved/rejected alerting the users.

Conclusion

Although our first project, we believe that this website along with its database is suitable for actually replacing the offline ticketing method, though the website might require minor changes and advancements for this. This has been a great learning opportunity for all of us, we learnt how to apply the concepts learnt in this course and also to develop a fully working website.

Though each of us have a specific part in the project assigned to us we coordinated the work among ourselves and helped each other during this quarantine period. We would like to thank Jyothi Kirar ma'am for understanding our problems regarding coordination among the team members and extending the deadline for submission of the project. We would also like to thank Ankur Tyagi sir for helping us in understanding with various SQL queries in our lab sessions, which helped us create a database for our project.

We sincerely thank Dr.Jyoti Kirar and Mr.Ankur Tyagi for providing us with such a wonderful opportunity.

We would like to conclude by saying that, we wish this project to be a first of many applications of all the concepts that we learnt in this course and beyond.

References

[1] W3 schools, https://www.w3schools.com/
[2] MySQL, https://www.mysql.com/
[3] PHP , https://www.php.net/
[4] XAMPP, https://xampp.site/
[5] HTML, https://html.com/
[6] *JavaScript, https://www.javascript.com/*