PREFACE

Excellence is an attitude that the whole of the human race is born with. It is the environment that makes sure that whether the result of this attitude is visible or otherwise. The well planned, properly executed and evaluated industrial training help a lot in including the good work culture. It provides linkage between student and industry in order to develop the awareness of industrial approach to problem solving based on broad understanding of process and mode of operation of an organization.

During this period, the students get there real first-hand experience on working in the actual environment. Most of the theoretical knowledge that they have gained during the course of their studies is put to test here. Apart from this, the students get opportunity to learn the latest technologies, which immensely help them in their career. This also benefits the organizations as many students doing their projects perform very well.

I had the opportunity to have the real practical experience, which has increased my sphere of knowledge to a great extent. Now I am better equipped to handle the real things than anyone else that has not undergone any such training. I learnt how an actual project progresses, what sort of problems actually occurs, how to produce quality products and so on. And being in such a reputed organization I had the best of experience.

TABLE OF CONTENTS

HISTORY OF PHP

PHP was conceived sometime in the fall of 1994 by Rasmus Lerdorf. Early non-released versions were used on his home page to keep track of who was looking at his online resume. The first version used by others was available sometime in early 1995 and was known as the Personal Home Page Tools. It consisted of a very simplistic parser engine that only understood a few special macros and a number of utilities that were in common use on home pages back then. A guestbook, a counter and some other stuff. The parser was rewritten in mid-1995 and named PHP/FI Version 2. The FI came from another package Rasmus had written which interpreted html form data. He combined the Personal Home Page tools scripts with the Form Interpreter and added mSQL support and PHP/FI was born. PHP/FI grew at an amazing pace and people started contributing code to it.

It is difficult to give any hard statistics, but it is estimated that by late 1996 PHP/FI was in use on at least 15,000 web sites around the world. By mid-1997 this number had grown to over 50,000. Mid-1997 also saw a change in the development of PHP. It changed from being Rasmus' own pet project that a handful of people had contributed to, to being a much more organized team effort. The parser was rewritten from scratch by Zeev Suraski and Andi Gutmans and this new parser formed the basis for PHP Version 3. A lot of the utility code from PHP/FI was ported over to PHP3 and a lot of it was completely rewritten.

INTRODUCTION TO PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.

- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

Abstract of the Project Bus Schedule and Ticket Booking System

The purpose of Bus Schedule and Ticket Booking System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Online Bus Schedule and Ticket Booking System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

The system also presents additional features including generating reports. These reports are included in the analysis and design of the system and also it emphasizes the importance of having this kind of a system. This system could be implemented easily among the people.

Introduction of the Project Bus Schedule and Ticket Booking System

Travel industry is evolving day to day. As the industry evolves, the need to digitalize all the transactions becomes need of the hour. A Software which provides an instant Solution for Bus Enquires, Ticket Booking and Cancelation of Tickets for a Customer is much Helpful.

Similarly, a Dealer needs a software to manage and maintain all the data of bus details, user's details, reservation details in one place.

The "Bus Schedule and Ticket Booking System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

This project helps to manage Bus scheduling for a Bus Dealer and provides a platform to maintain all the data. Tickets can be booked by a Customer and Cancelled before the bus departure. This software can be used to search bus routes from specific location to its destination and book them online.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Bus Ticket Booking System, as described above, can lead to error free,

secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Tickets, Bus, Customer, Bus Schedule, Bus Route. Every Online Bus Ticket Booking System has different Bus needs; therefore, we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

Objective of Project on Bus Schedule and Ticket Booking System:

The main objective of the Project on Bus Schedule Ticket Booking System is to manage the details of Bus, Ticket, Booking, Customer, Bus Route. It manages all the information about Bus, Bus Schedule, Bus Route, Bus. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work

for managing the Bus, Ticket, Bus Schedule, Booking. It tracks all the details about the Booking, Customer, Bus Route. The main objective is to Develop the System with a user-friendly interface and easy procedures to attract Passengers. Using easy Payment methods and Implementing solutions through System and making it more effective such as developing system to Book Bus Ticket form anywhere in the World.

Functionalities provided by Bus Ticket Booking System are as follows:

- Provides the searching facilities based on various factors. Such as Bus Source and Destination place, Bus Time and Date and Bus Route.
- Bus Ticket Booking System also manage the Bus Schedule details and Customer details, Bus Route details, User details and Ticket details.
- It tracks all the information of Ticket, Bus Schedule, Customer etc.
- Manage the information of Tickets
- Shows the information and description of the Bus, City and Customer.
- To increase efficiency of managing the Bus, Route.
- It deals with monitoring the information and transactions of Booking.
- Manage the information of Bus
- Editing, adding and updating of Records is improved which results in proper resource management of Bus data.
- Manage the information of Booking.
- Manage the information of Customer.
- Integration of all records of Customer.

Existing System

Existing System refers to the system that is being followed till now. The Existing System requires more Computational Time, more Manual Calculations, and the complexity involved in Selection of features is high. The other disadvantages are lack of Security of Data, Deficiency of Data accuracy, Time consuming etc. To avoid all these limitations and make the working more accurately the System needs to be Computerized.

The Existing System is not completely Computerized. The Customer has to visit any booking branch if he wants to book a Ticket. Bus scheduling, Ticket booking, Bill generation and many other operations are done manually. This may lead to incorrect entries and there is a lot of room for errors as the data is not completely synced. The availability of seats is not centrally maintained and the travel operator is not fully aware of the availability and occupancy of the seats in his Buses. This is the major bottle neck. To avoid all these limitations and make the working more accurately the System needs to be Computerized.

Proposed System

The aim of Proposed System is to develop a System of improved facilities. The Proposed System can overcome all the limitations of the Existing System. The System provides proper Security and reduces the Manual work. The Existing System has several Disadvantages and many more difficulties to work well. The Proposed System tries to eliminate and reduce these difficulties up to some extent. The Proposed System will help the user to reduce the workload. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

Bus scheduling and booking system is a web-based application which is connected to a complete database. The database includes information about buses, number of seats available, occupancy, availability, date and time of Bus Operation, Source Station and Destination Station, Price of the Ticket, automated report and Bill generation etc. User can also cancel their ticket from anywhere and anytime. The database is updated time to time and the application gives utmost importance to security and usability.

Scope of the project Bus Schedule and Ticket Booking System:

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Bus Schedule and Ticket Booking System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Bus Schedule and Ticket Booking System.

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfies the user requirement
- It is easy to understand by the user and operator
- It is easy to operate
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

Modules of Bus Schedule and Ticket Booking System:

- **Add Bus Module:** Used for Adding or Creating a new Bus and its Details.
- **Update Bus Module:** Used for Updating or Modifying the details of existing Bus.
- **Bus Schedule Module:** Used for managing the details of Bus Scheduling.
- **Ticket Management Module:** Used for managing the information and details of the Ticket.
- **Bus Searching Module:** Used for Bus searching process.
- **Booking Module:** Used for managing the Booking details of a passenger.
- **Customer Module:** Used for managing the Customer information in his/her profile section.
- **Registration Module:** Used for managing the Registration process.
- **Login Module:** Used for managing the login details of customer and admin.
- **Admin Module:** Used for managing the functions of the admin of the system.
- **Cancelation Module:** Used for Cancelation process of Tickets.
- **Payment Module:** Used to make Payment.

Minimum Hardware Requirement:

Client Side Requirement :	
Processor :	1.5GHz
RAM :	512 MB
Hard-Disk :	80 GB

Server Side Requirement :	
Processor :	2.5GHz
RAM :	1 GB
Hard-Disk :	80 GB

Minimum Software Requirement:

Client Side Requirement :	
Operating System :	Windows 2007/XP or equivalent OS.
Web Browser :	Internet Explorer 6.0 or any compatible Browser.

Server Side Requirement :	
Front-End :	PHP 5.3.0
Back-End :	MYSQL 5.1.36
Web Server :	WAMP 3.1

Input Data and Validation of Project on Bus Scheduleand Ticket Booking System:

- All the fields such as Bus, City, Customer are validated and does not take invalid values.
- Each form for Bus, Route, Registration cannot accept blank value fields.
- Avoiding errors in data.
- Controlling amount of input.
- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.
- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.
- Testing the module with all the possible test data.
- Testing of the functionality involving all type of calculations etc.
- Commenting standard in the source files.

Features of the project Bus Ticket Booking System:

- Product and Component based
- Creating & Changing Issues at ease
- Query Issue List to any depth
- Reporting & Charting in more comprehensive way
- User Accounts to control the access and maintain security
- Simple Status & Resolutions
- Multi-level Priorities & Severities.
- Targets & Milestones for guiding the programmers
- Attachments & Additional Comments for more information
- Robust database back-end
- Various level of reports available with a lot of filter criteria's
- It contains better storage capacity.
- Accuracy in work.
- Easy & fast retrieval of information.
- Well-designed reports.
- Decrease the load of the person involve in existing manual system.
- Access of any information individually.
- Work becomes very speedy.
- Easy to update information

Software Requirement Specification:

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

The proposed system has the following requirements

- System needs store information about new entry of Bus.
- System needs to help the internal staff to keep information of Ticket and find them as per various queries.
- System need to maintain quantity record.
- System need to keep the record of City.
- System need to update and delete the record.
- System also needs a search area.
- It also needs a security system to prevent data.

Identification of need:

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating

any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business. For this reason, we have provided features in Present system is partially automated (computerized), actually existing system is quite laborious as one has to enter same information at three different places.

System Design of Bus Ticket Booking System

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. **Primary Design Phase:**

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2. **Secondary Design Phase:**

In the secondary phase the detailed design of every block is performed.

The general tasks involved in the design process are the following:

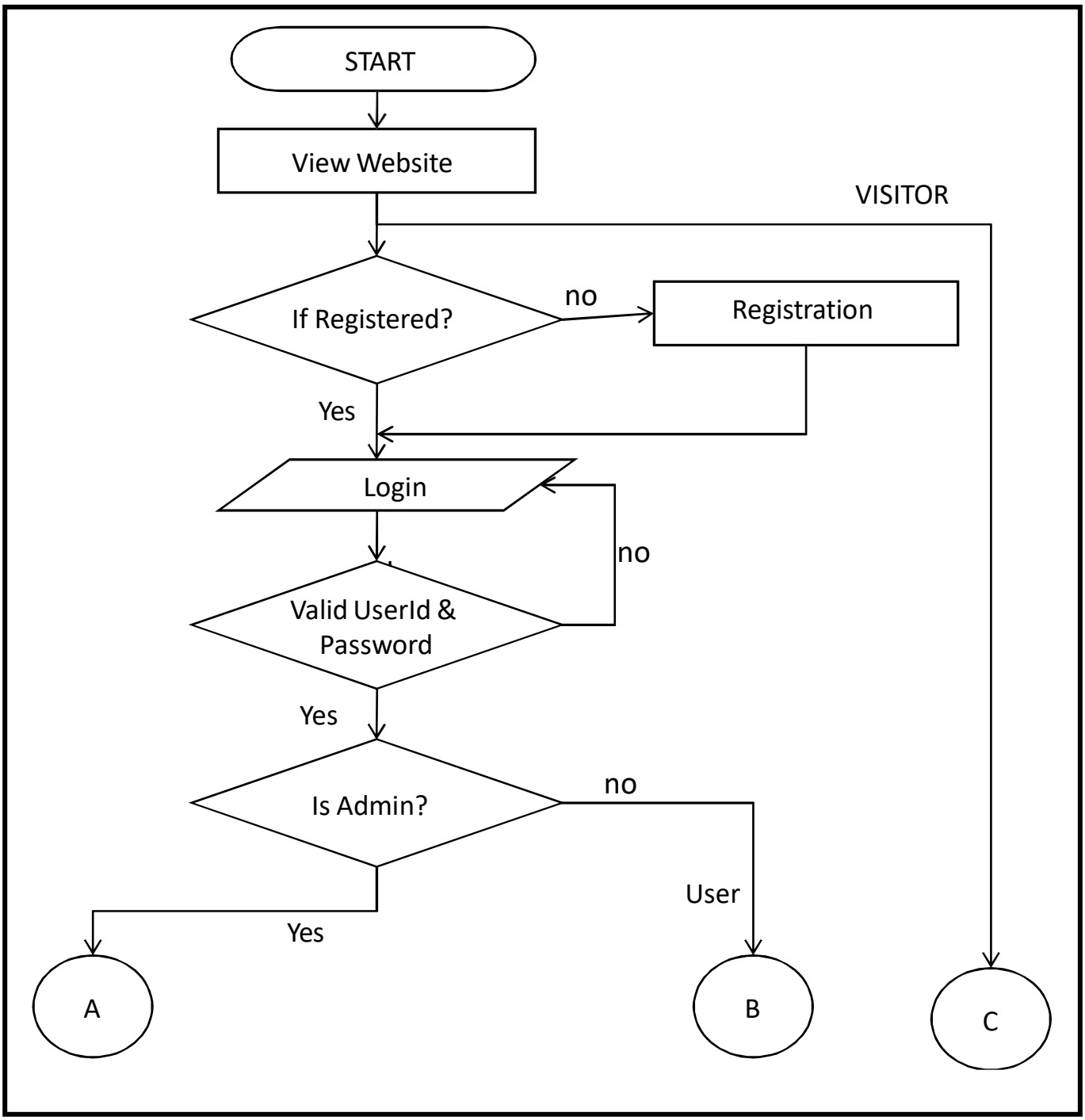
1. Design various blocks for overall system processes.
2. Design smaller, compact and workable modules in each block.
3. Design various database structures.
4. Specify details of programs to achieve desired functionality.
5. Design the form of inputs, and outputs of the system.
6. Perform documentation of the design.
7. System reviews.

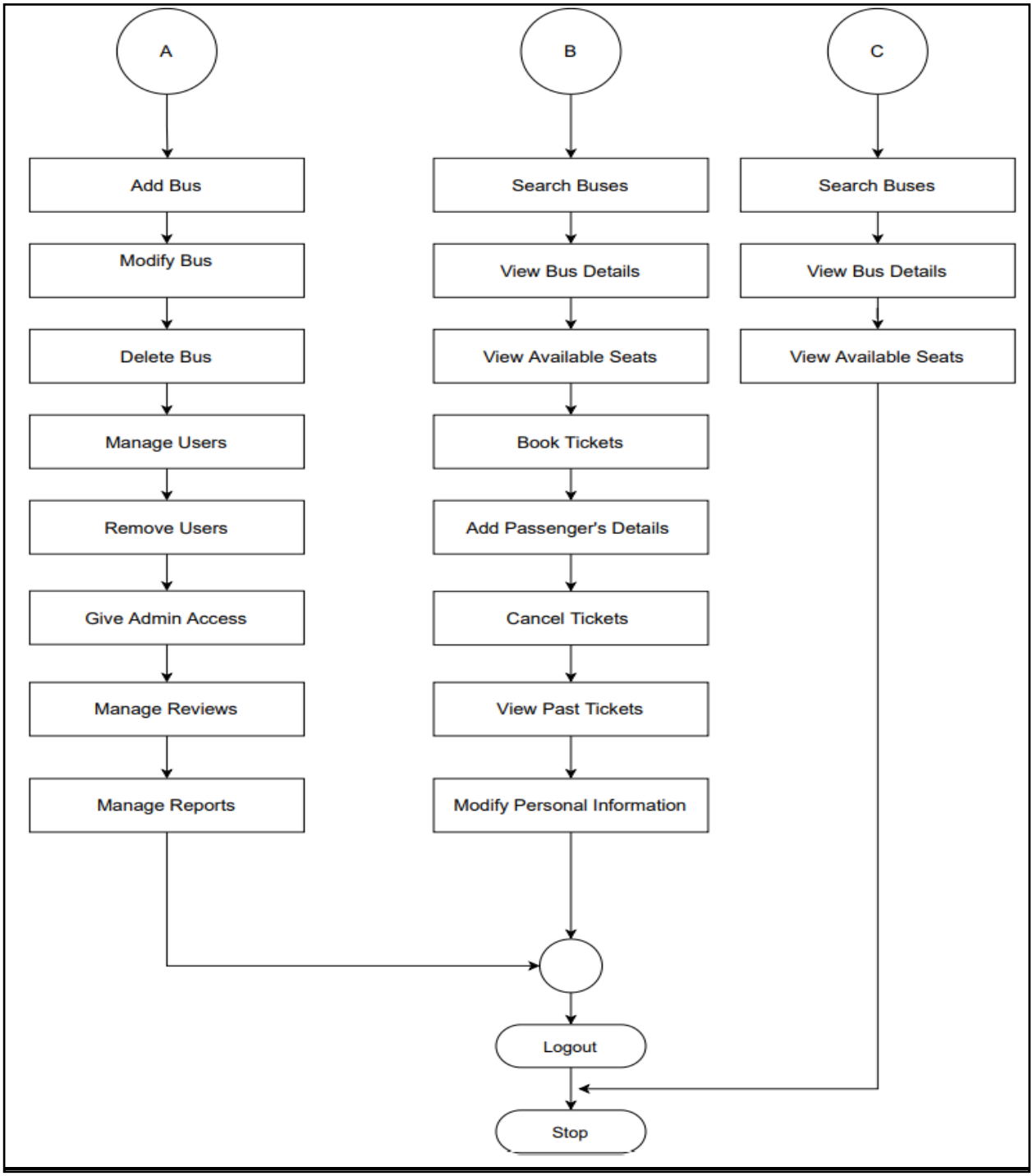
Dataflow Diagram:

Data flow diagram is the starting point of the design phase that functionally decomposes the requirements specification. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. A DFD describes what data flow rather than how they are processed, so it does not hardware, software and data structure.

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model. The data flow diagram is a graphical description of a system's data and how to Process transform the data is known as Data Flow Diagram (DFD).

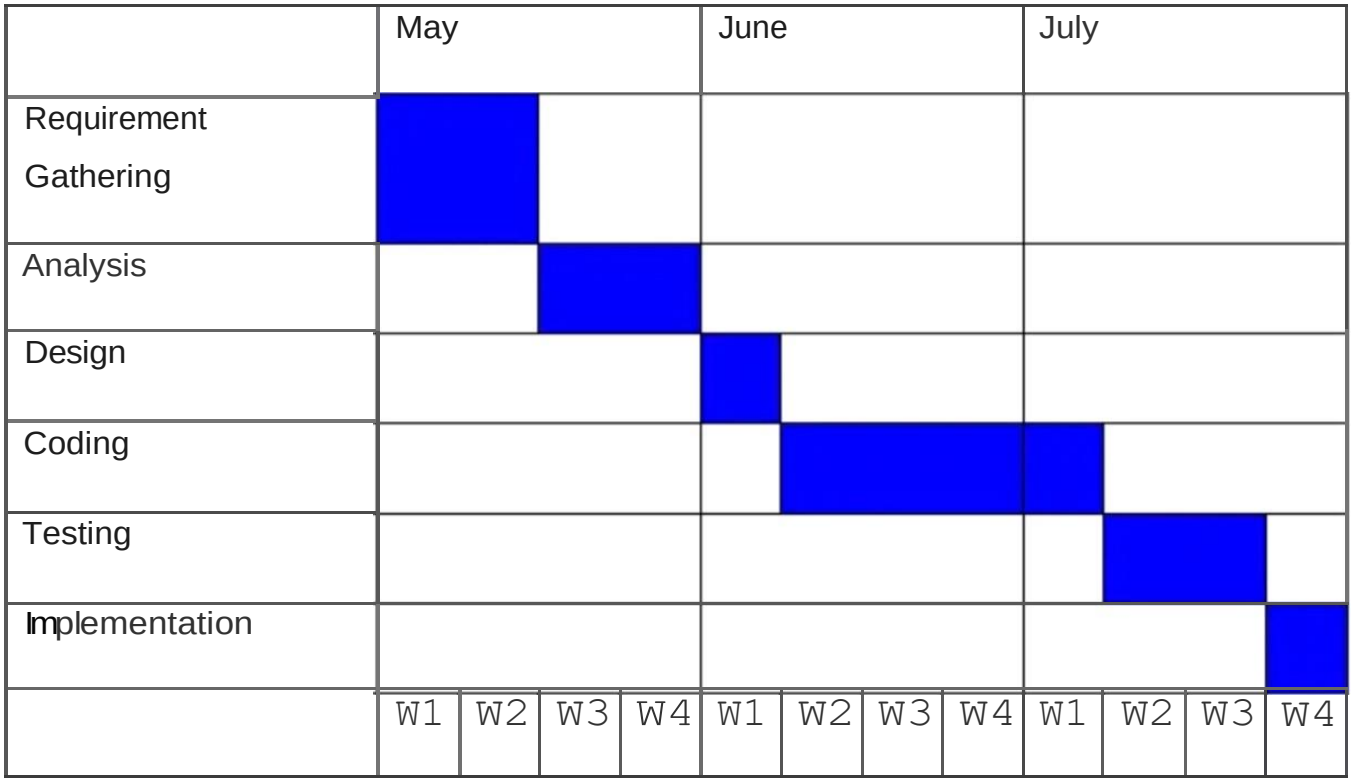
System flow Chart of Bus Schedule and Ticket Booking System




A
B
C
Add Bus
Modify Bus
Delete Bus
Manage Users
Remove Users
Give Admin Access
Manage Reviews
Manage Reports
Search Buses
View Bus Details
View Available Seats
Book Tickets
Add Passenger's Details
Cancel Tickets
View Past Tickets
Modify Personal Information
Search Buses
View Bus Details
View Available Seats
Logout
Stop

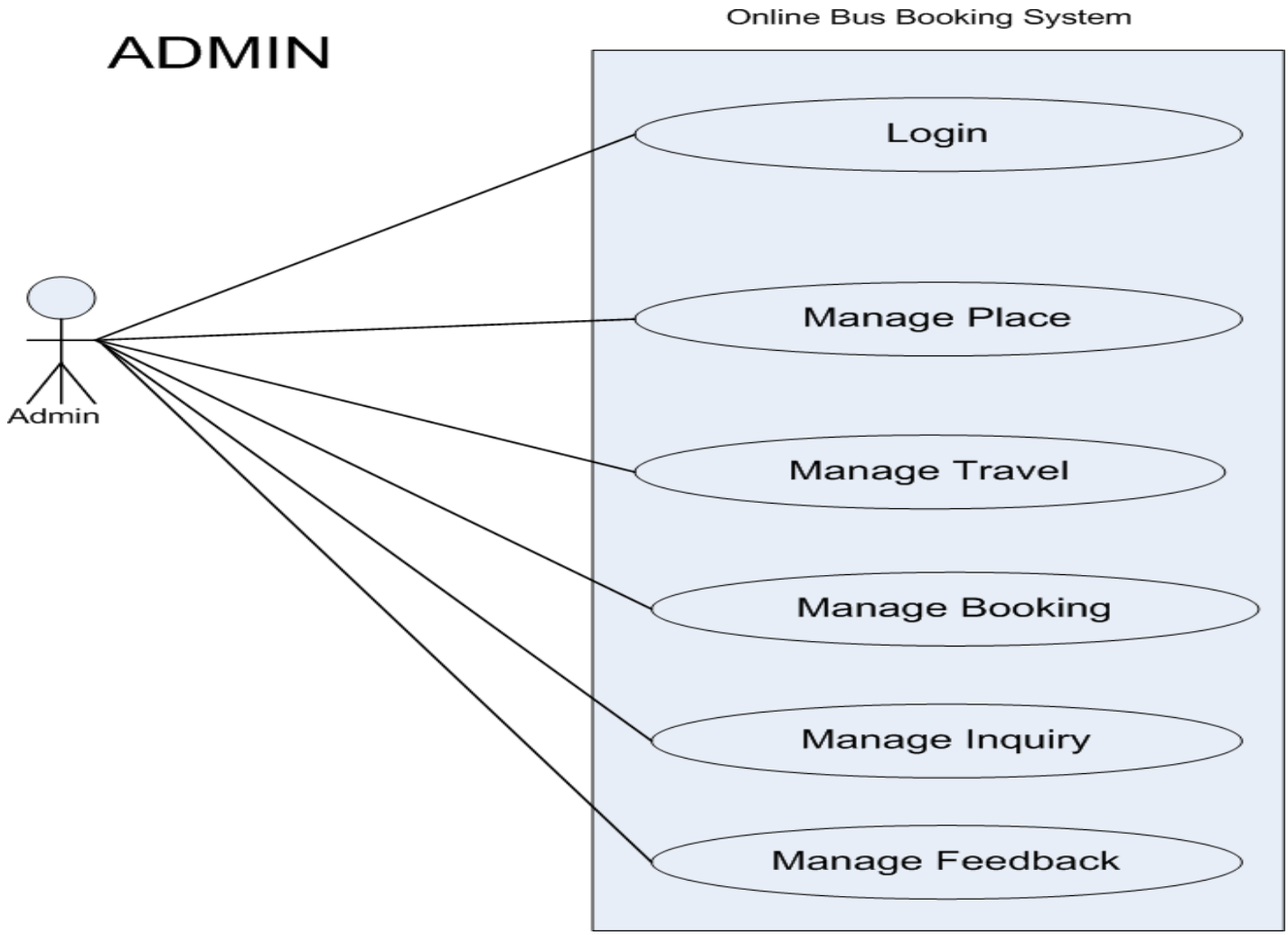
Project Scheduling:

An elementary Gantt chart or Timeline chart for the development plan is given below. The plan explains the tasks versus the time (in weeks) they will take to complete

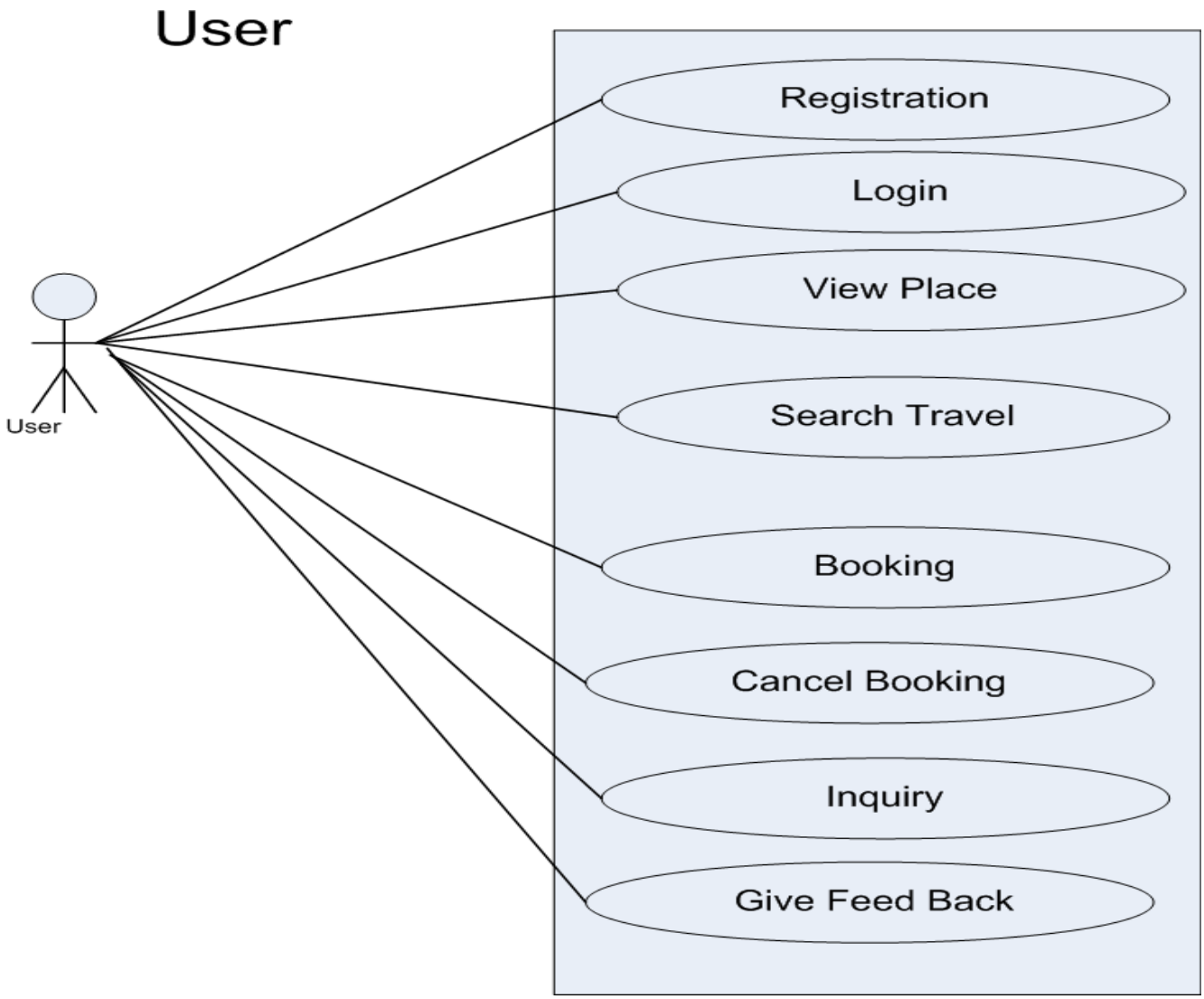


[W1-W4 = weeks in the month]

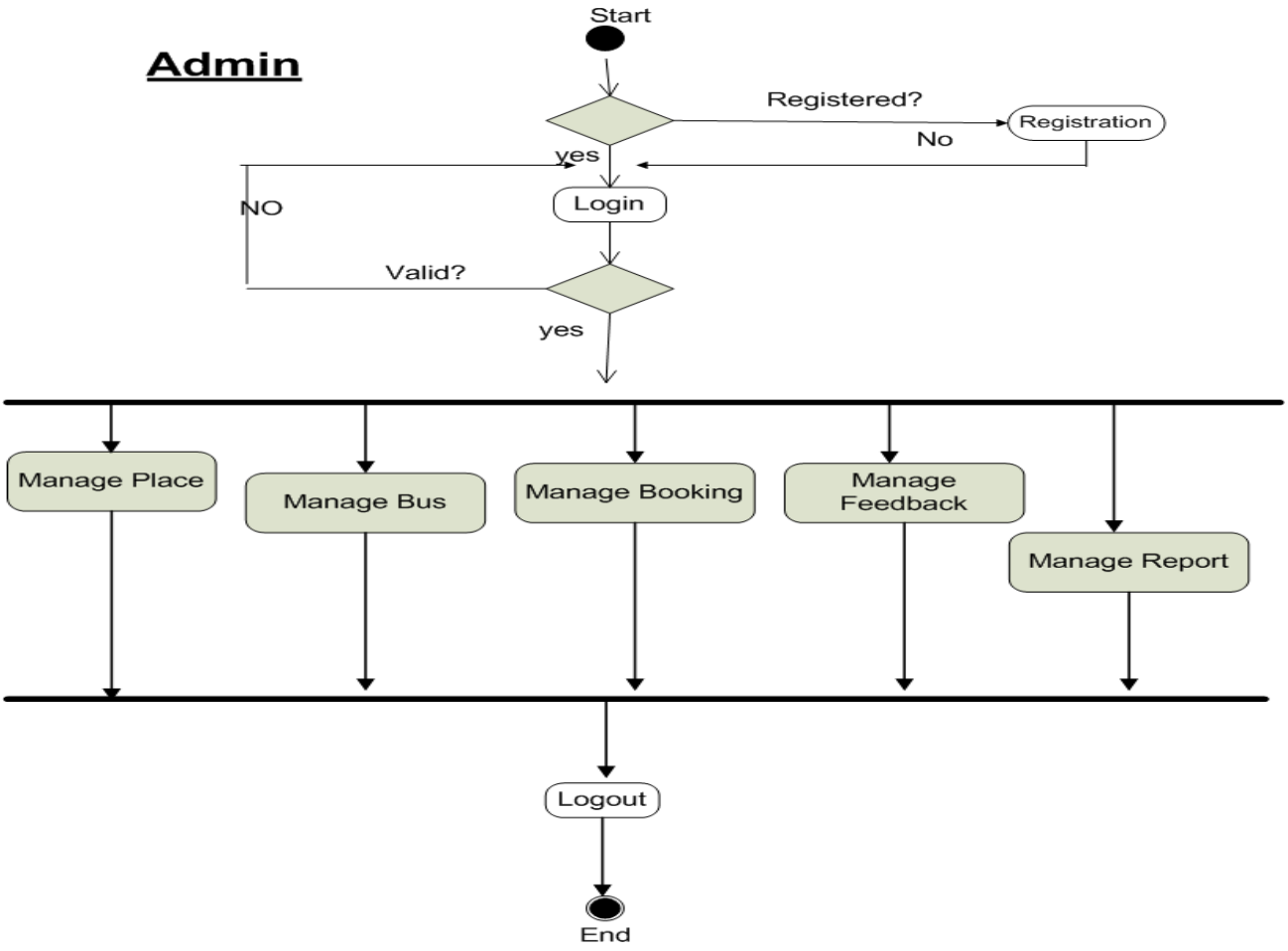
Use-Case Diagram for Admin



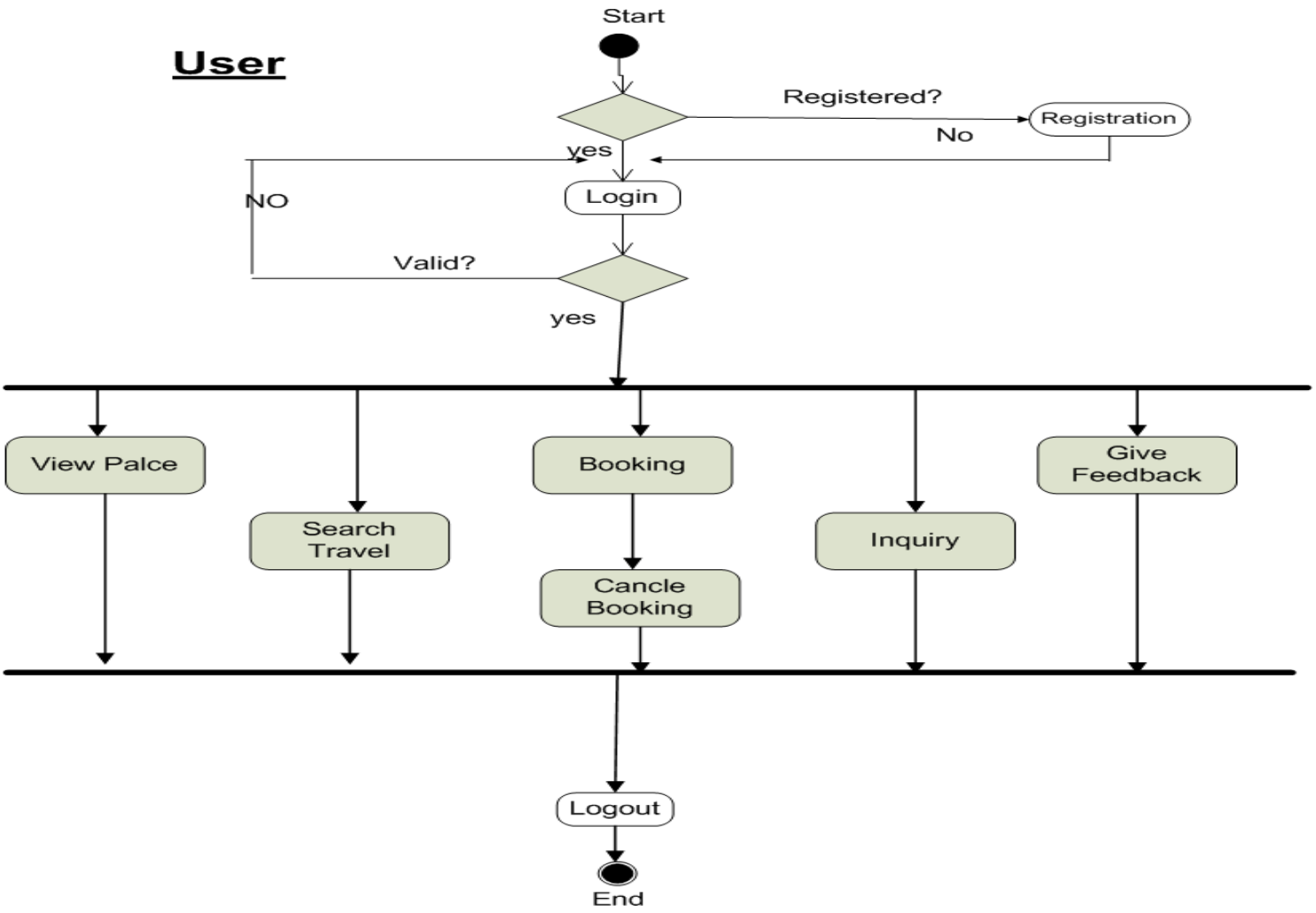
Use-Case Diagram for User



Activity Diagram for Admin



Activity Diagram for User



Data Dictionary:

This is normally represented as the data about data. It is also termed as metadata some times which gives the data stored in the database. It defines each data term encountered during the analysis and of a new system. Data elements can describe files or the processes.

Database and Table Structure of the Project

Table Name – User

No	Field Name	Data type	Constraint	Description
1	user_id *(Primary)*	int(3)	Primary Key	ID
2	username	varchar(255)	Not Null	User Name
3	user_password	varchar(255)	Not Null	Password
4	user_firstname	varchar(255)	Not Null	Firstname
5	user_lastname	varchar(255)	Not Null	Lastname
6	user_email	varchar(255)	Not Null	Email
7	user_phoneno	varchar(255)	Not Null	Phone number
8	user_image	text	Not Null	User Image
9	user_role	varchar(255)	Not Null	User Role

Table Name – Posts [New Bus]

No	Field Name	Data type	Constraint	Description
1	post_id *(Primary)*	int(3)	Primary Key	new bus id
2	post_category_id	int(3)	Not Null	new bus category
3	post_title	varchar(255)	Not Null	new bus title
4	post_author	varchar(255)	Not Null	new bus author
5	post_date	date	Not Null	new bus date
6	post_image	text	Not Null	new bus image
7	post_content	text	Not Null	new bus content
8	post_source	varchar(255)	Not Null	new bus source
9	post_destination	varchar(255)	Not Null	new bus destination
10	post_via	varchar(255)	Not Null	intermediate stations
11	post_via_time	varchar(255)	Not Null	new bus time
12	post_query_count	int(3)	Not Null	new bus review
13	max_seats	int(3)	Not Null	new bus seats
14	available_seats	int(3)	Not Null	new bus seats
15	seat_cost	int(4)	Not Null	Ticket Cost

Table Name – Orders

No	Field Name	Data type	Constraint	Description
1	order_id *(Primary)*	int(3)	Primary Key	Order Id
2	bus_id	int(3)	Not Null	Bus Id
3	user_id	int(3)	Not Null	User Id
4	user_name	varchar(255)	Not Null	User Name
5	user_age	int(3)	Not Null	User Age
6	source	varchar(255)	Not Null	Source Station
7	destination	varchar(255)	Not Null	Destination Station
8	date	date	Not Null	Date
9	cost	int(3)	Not Null	Ticket Cost
10	Payment	varchar(16)	Not Null	Payment

Table Name – Seats

No	Field Name	Data type	Constraint	Description
1	bus_id *(Primary)*	int(3)	Primary Key	Bus Id
2	max_seats	int(3)	Not Null	Maximum Seats
3	available_seats	int(3)	Not Null	Available Seats

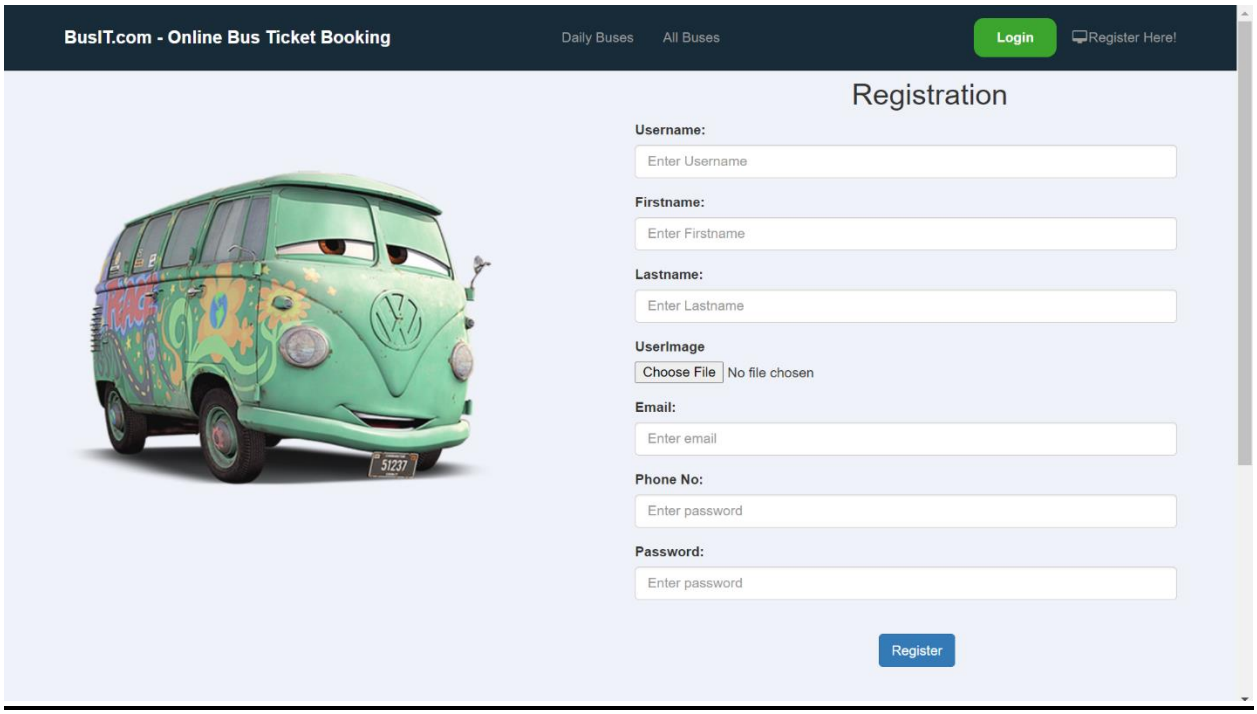
Table Name – Query [Reviews]

No	Field Name	Data type	Constraint	Description
1	query_id *(Primary)*	int(3)	Primary Key	Review Id
2	query_bus_id	int(3)	Not Null	Review Bus Id
3	query_user	varchar(255)	Not Null	User Name
4	query_email	varchar(255)	Not Null	Email
5	query_date	date	Not Null	Date
6	query_content	text	Not Null	Review Content
7	query_replied	varchar(255)	Not Null	Replied or not

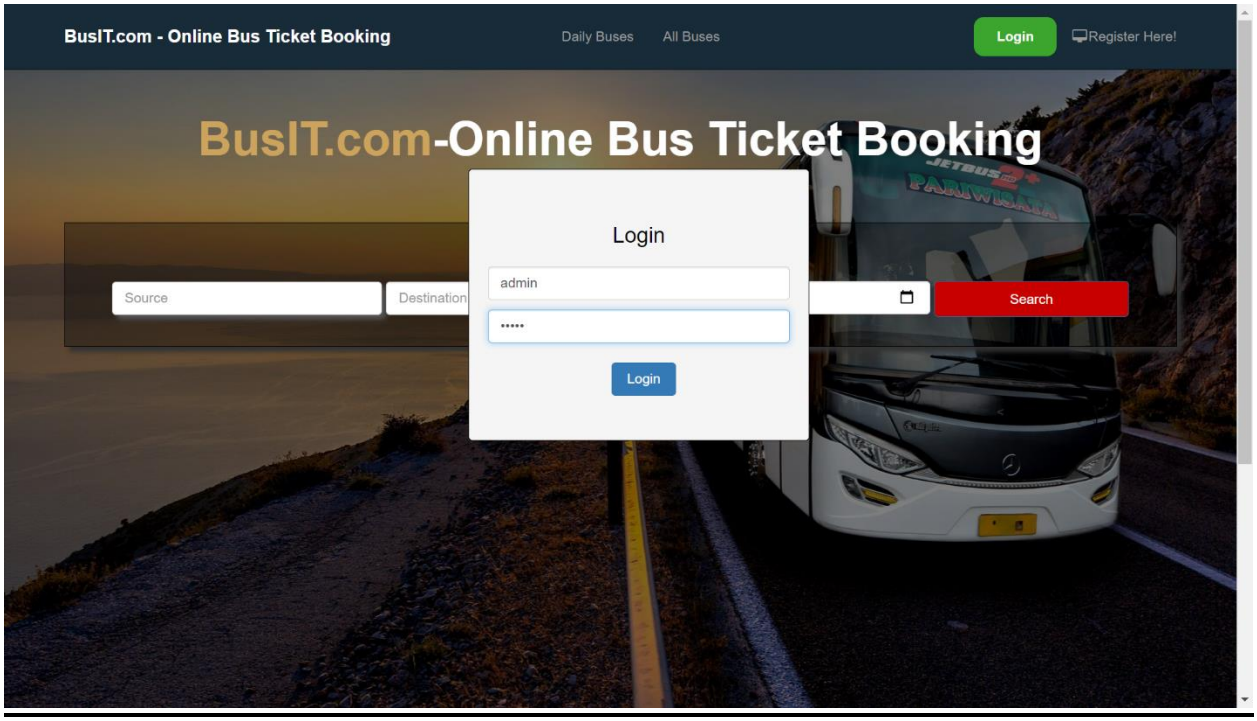
Input Output Design

Home Page

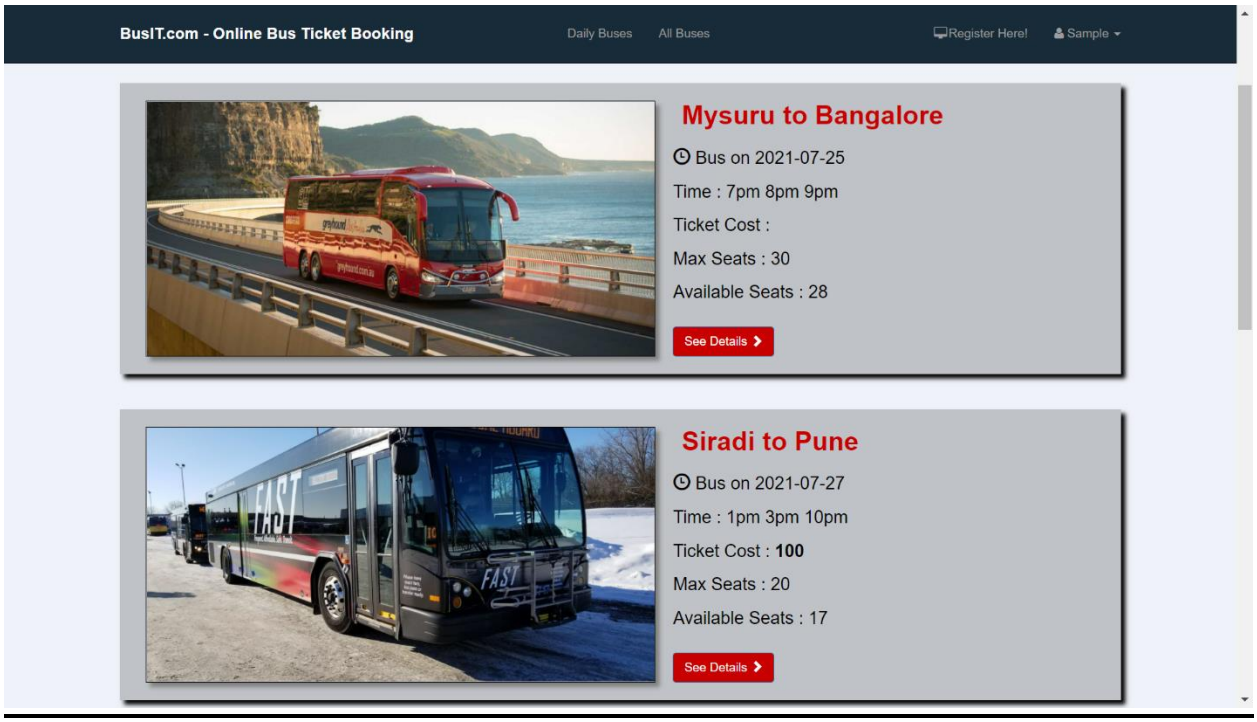
Registration Page



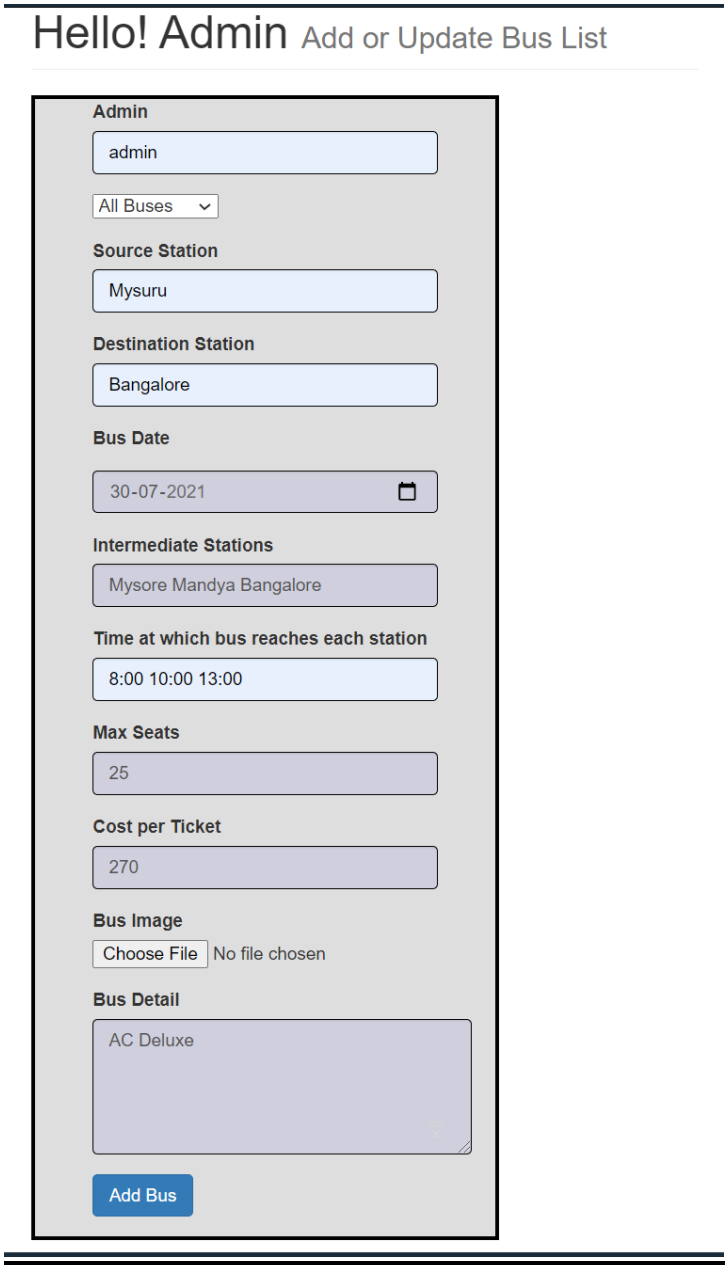
Login Page



Available Bus List Page

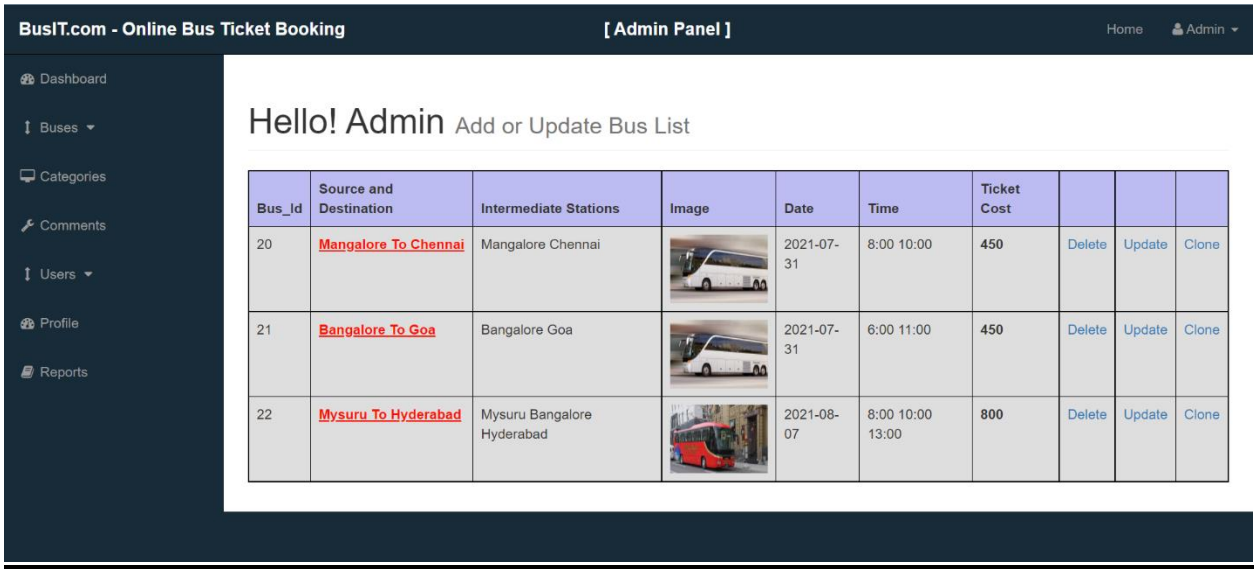


Add Bus [Admin] Page

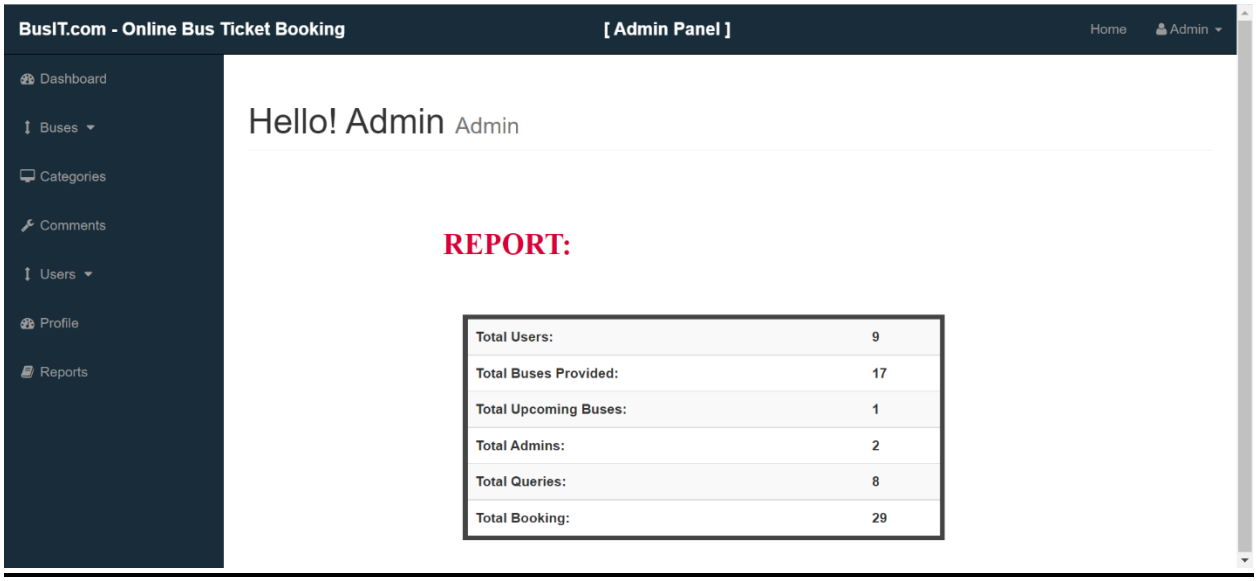


Search Bus Bar

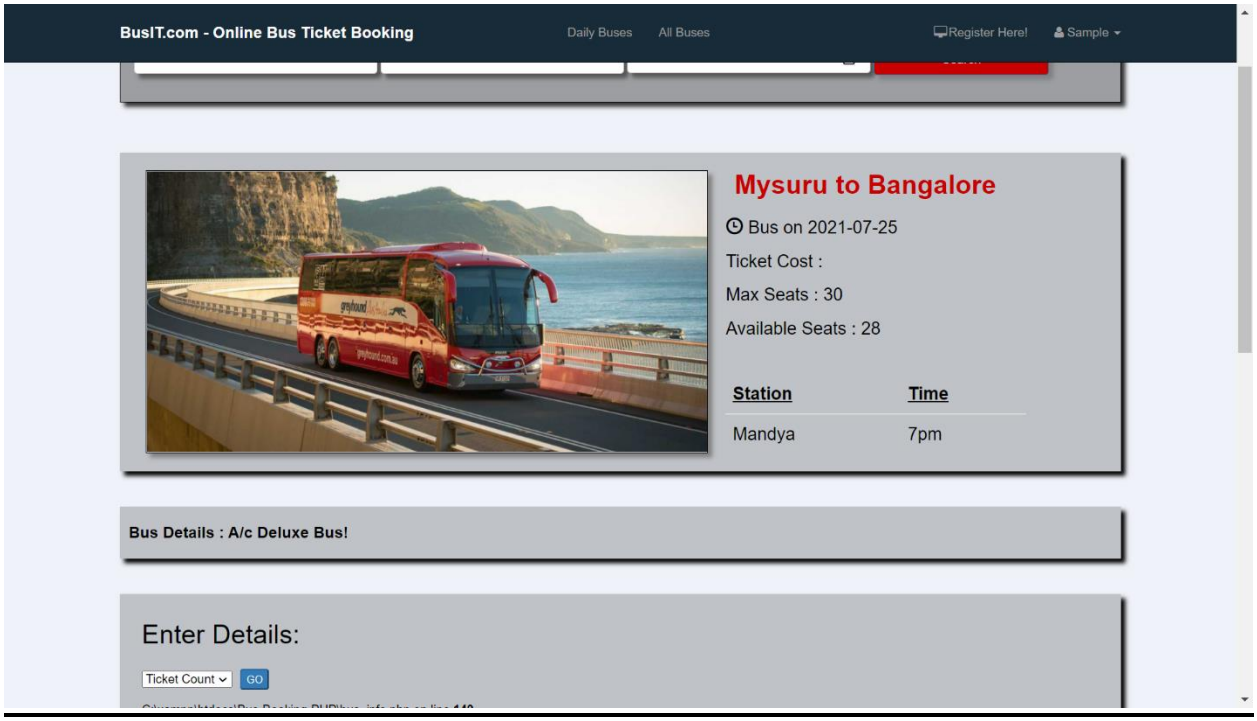
Admin Bus List page

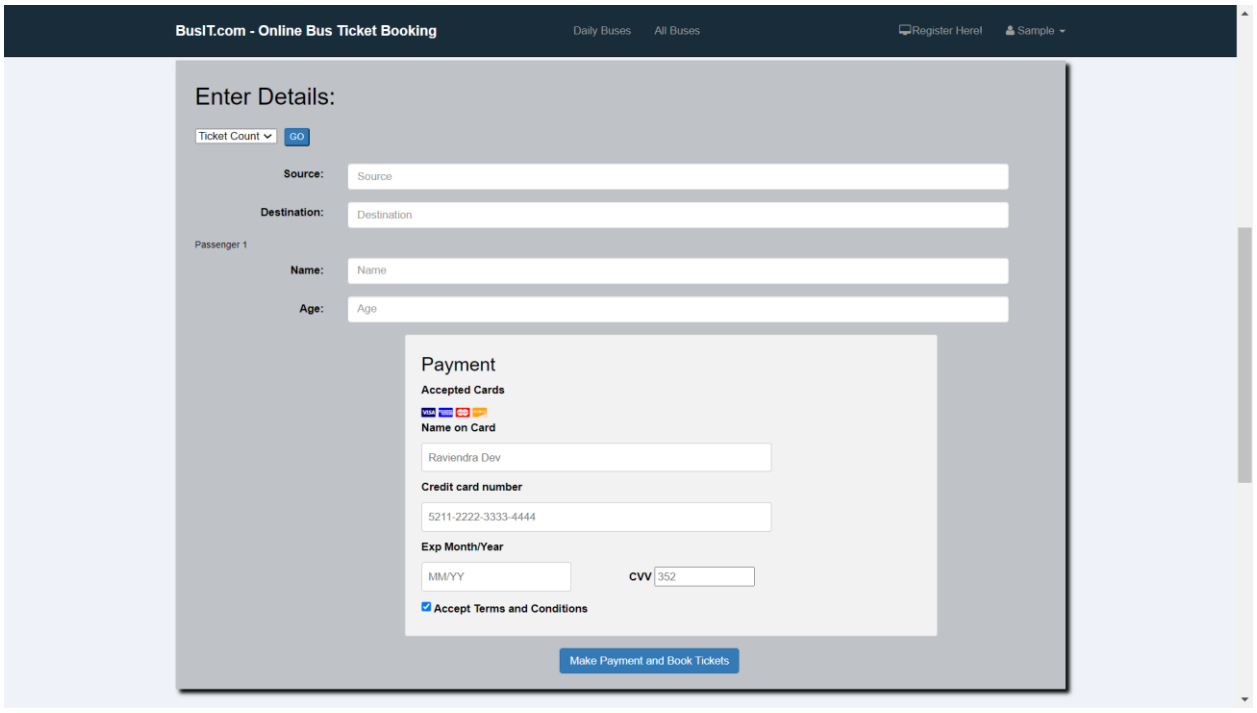


Admin Report Page

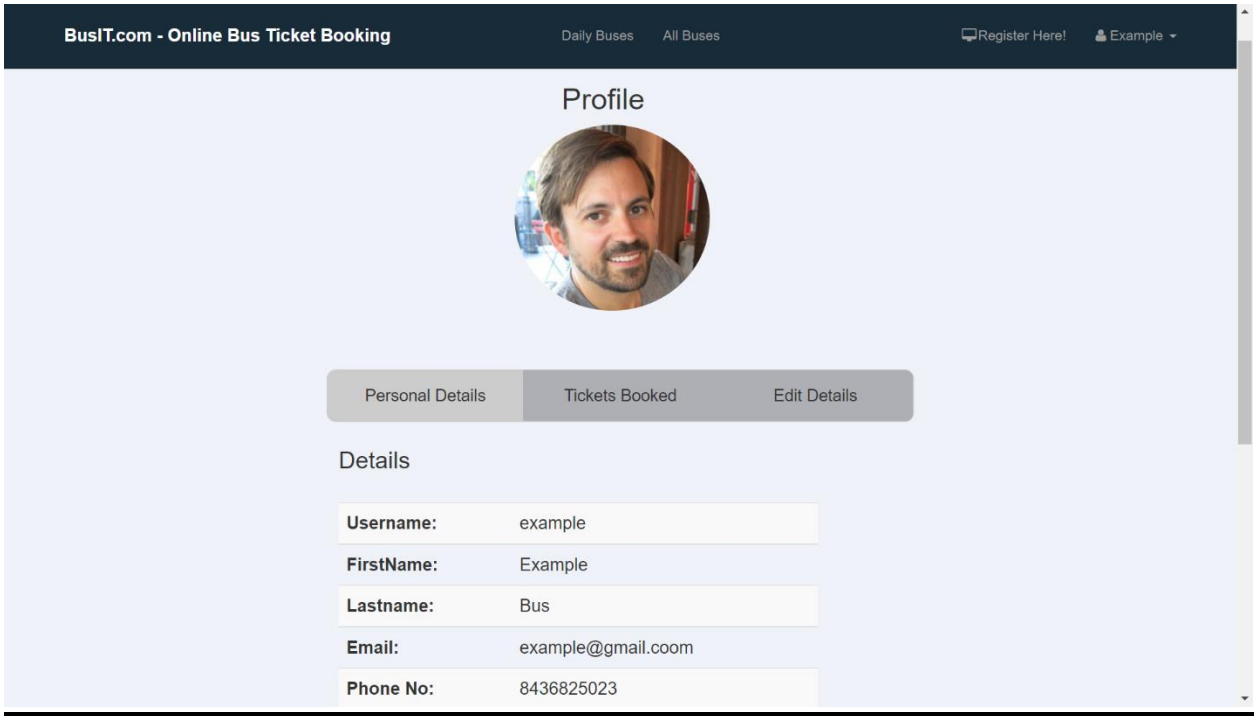


Bus Info Page and Ticket Booking Page

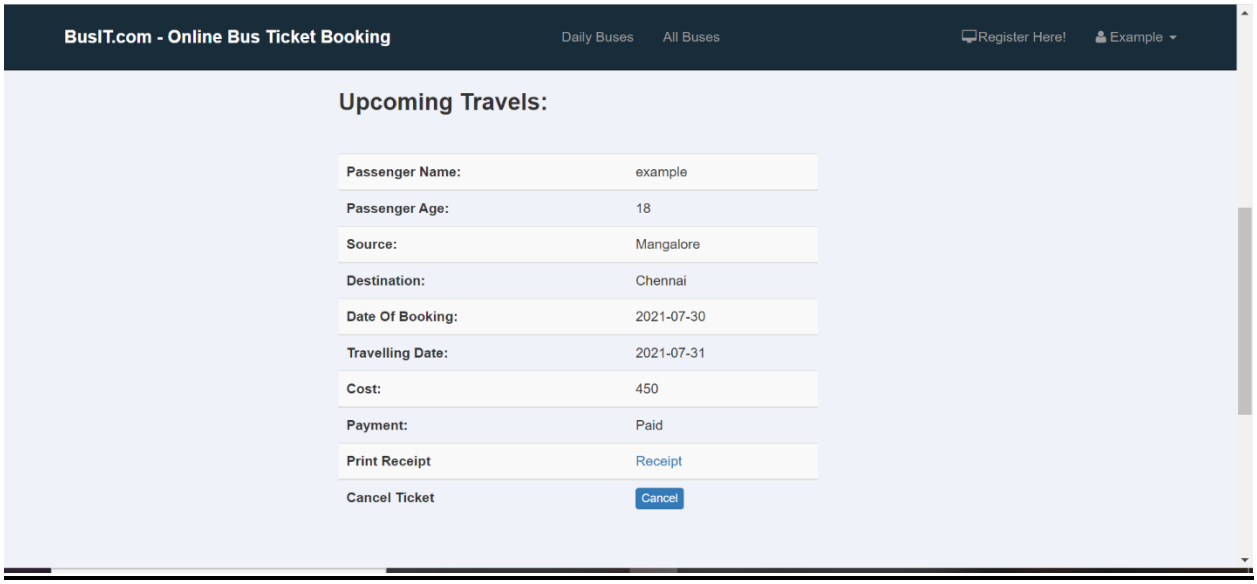




Profile Page



Ticket Info and Cancelation Page



Coding Of the Project:

Search Bus

```
<div id="bus_search_bar" class="container bus_search" >
    <h4 style="color:#fff;"><b>Bus Search</b></h4>
       <form class="bus_form" action="search.php" method="post">
        <input id="form-item" name="source" type="text" class="form-control" placeholder="Source">
        <input id="form-item" name="destination" type="text" class="form-control" placeholder="Destination">
        <input id="form-item" type="date"  min=<?php echo date('Y-m-d');?> max=<?php echo date('Y-m-d', strtotime(date('Y-m-d'). '+ 29 days'));?> name="date" class="form-control" id="date" placeholder="dd/mm/yyyy" >

        <button class="btn btn-primary" name="submit">Search</button>
        </form>
  </div>
```

Adding New Bus

```
<?php
        if (isset($_POST['insert-bus'])) {
                $admin = $_POST['admin'];
                $category = $_POST['category'];
                $source = $_POST['source'];
                $destination = $_POST['destination'];
                $title = $source . " to " . $destination;
                $intermediate = $_POST['intermediate'];
                $date = $_POST['date'];
                $via_time = $_POST['via-time'];
                $bus_detail = $_POST['bus-detail'];
                $max_seats = $_POST['max_seats'];
                $seat_cost = $_POST['seat_cost'];

                $image = $_FILES['image']['name'];
                $tmp_image = $_FILES['image']['tmp_name'];
                move_uploaded_file($tmp_image, "images/$image");

                if ($admin=="" || $category=="" || $source=="" || $destination=="" || $title=="" || $intermediate=="" || $date=="" || $via_time=="" || $bus_detail=="" || $max_seats=="" || $seat_cost=="") { echo "**All Fields Mandatory";}
                else {
                        $query = "INSERT INTO posts(post_category_id, post_title, post_author, post_date, post_image, post_content, post_source, post_destination, post_via, post_via_time, max_seats, seat_cost, available_seats)
```

```
                        VALUES({$category}, '{$title}', '{$admin}', '{$date}', '{$image}',
'{$bus_detail}', '{$source}', '{$destination}', '{$intermediate}', '{$via_time}', $max_seats,
$seat_cost, $max_seats)";
                        $bus_entry = mysqli_query($connection,$query);
                        if (!$bus_entry) {
                                die("Query Failed");
                        }
                }
        }
?>
```

Update Bus Details

```
<?php

if (isset($_GET['bus_id'])) {
        $edit_bus_id = $_GET['bus_id'];
}

$query = "SELECT *  FROM  posts WHERE post_id=$edit_bus_id";
$select_posts = mysqli_query($connection,$query);

while($row = mysqli_fetch_assoc($select_posts)) {
   $bus_id = $row['post_id'];
   $admin_name = $row['post_author'];
   $source = $row['post_source'];
   $destination = $row['post_destination'];
   $intermediate_station = $row['post_via'];
   $category = $row['post_category_id'];
   $detail = $row['post_content'];
   $image = $row['post_image'];
   $date = $row['post_date'];
                $seat_cost = $row['seat_cost'];
   $time = $row['post_via_time'];
}

if (isset($_POST['update-bus'])) {

        $admin = $_POST['admin'];
        $category = $_POST['category'];
        $source = $_POST['source'];
        $destination = $_POST['destination'];
        $title = $source . " to " . $destination;
        $intermediate = $_POST['intermediate'];
        $date = $_POST['date'];
        $via_time = $_POST['via-time'];
        $bus_detail = $_POST['bus-detail'];
        $seat_cost = $_POST['seat_cost'];

        $query = "UPDATE posts SET post_title='{$title}', post_date='{$date}',
post_source='{$source}', post_destination='{$destination}', post_author='{$admin}',
```

```
post_category_id={$category}, post_via='{$intermediate}', post_via_time='{$via_time}',
seat_cost='{$seat_cost}', post_content='{$bus_detail}'
        WHERE post_id=$edit_bus_id ";

        $update_bus = mysqli_query($connection,$query);
        if (!$update_bus) {
                die("Query Failed" . mysqli_error($connection));
        }
}
?>
```

Registration:

```
<?php

if (isset($_POST['register'])) {
    $username = $_POST['username'];
    $firstname = $_POST['firstname'];
    $lastname = $_POST['lastname'];
    $email = $_POST['email'];
    $phone_no = $_POST['phone_no'];
    $password = $_POST['password'];

    $image = $_FILES['image']['name'];
    $tmp_image = $_FILES['image']['tmp_name'];

    move_uploaded_file($tmp_image, "images/$image");
    if ($image == "") {
      $image = "user_default.jpg";
    }

if ($username=="" || $firstname=="" || $lastname=="" || $email=="" || $phone_no=="" ||
$image=="" || $password=="") {
  echo "**ALL FIELDS MANDATORY";
}
elseif (strlen($phone_no)!=10) {
  echo "**PhoneNo Must Contain Of 10 bits";
}
else {

$query = "INSERT INTO users(username, user_password, user_firstname, user_lastname,
user_email, user_phoneno, user_role, user_image) VALUES('$username', '$password',
'$firstname', '$lastname', '$email', '$phone_no', 'subscriber', '$image') ";
$register_user = mysqli_query($connection, $query);
if(!$register_user) {
    die("Query Failed" . mysqli_error($connection));
}
header("Location: index.php");
}
}?>
```

Login

```
<?php include "db.php"; ob_start(); ?>
<?php session_start(); ?>
<?php
if (isset($_POST['login']) || isset($_POST['register'])) {
	$username = $_POST['username'];
	$password = $_POST['password'];

	$query = "SELECT * FROM users WHERE username = '$username'";
	$select_user = mysqli_query($connection,$query);

	if (!$select_user) {
		die("Query Failed" . mysqli_error($connection));
	}

	while ($row = mysqli_fetch_assoc($select_user)) {
		$db_user_id = $row['user_id'];
		$db_username = $row['username'];
		$db_user_password = $row['user_password'];
		$db_user_firstname = $row['user_firstname'];
		$db_user_lastname = $row['user_lastname'];
		$db_user_role = $row['user_role'];
		$db_user_image = $row['user_image'];

		if($username === $db_username && $password === $db_user_password) {

			$_SESSION['s_username'] = $db_username;
			$_SESSION['s_firstname'] = $db_user_firstname;
			$_SESSION['s_lastname'] = $db_user_lastname;
			$_SESSION['s_role'] = $db_user_role;
			$_SESSION['s_image'] = $db_user_image;
			$_SESSION['s_id'] = $db_user_id;
			if ($db_user_role == 'admin') {
				header("Location: ../admin");
				exit;
			}
			else if ($db_user_role == 'subscriber') {
				header("Location: ../index.php");
				exit;
			}}
		else {
			header("Location: ../index.php");
			exit;}
	}
}?>
```

Database

```
<?php

$connection = mysqli_connect("localhost",'root',",'bus');

if(!$connection) {
        die("Unable to connect" . mysqli_error($connection));
}
?>
```

Logout

```
<?php session_start(); ?>
<?php

$_SESSION['s_username'] = null;
$_SESSION['s_firstname'] = null;
$_SESSION['s_lastname'] = null;
$_SESSION['s_role'] = null;

header("Location: ../index.php");

?>
```

Ticket Booking

```
if (isset($_SESSION['s_id'])) {
          ?>
     <div class="col-lg-12" style="background-color: rgba(0, 0, 0, 0.20); color: #000; margin-
bottom: 40px; padding:10px;box-shadow:5px 5px 5px;">
            <div class="container">
<h2>Enter Details:</h2>

        <form action="" method="post" class="form-horizontal">

        <select name="passenger_count" style="margin-bottom: 15px;margin-top: 15px;">
        <option value="0">Ticket Count</option>
        <?php
                for ($i=1; $i <= $sum; $i++) { ?>
                <option value="<?php echo $i ?>"><?php echo $i ?></option> <?php
                        }?>
        </select>
       <button class="btn-xs btn-primary" style="margin-left: 5px;">GO</button>

        </form>
        <form action="bus_info.php?bus_id=<?php echo $selected_bus ?>&count=<?php
        echo $_POST['passenger_count'] ?>" method="post" class="form-horizontal">
                <div style="padding-top:10px"class="form-group">
```

```
                <label style="padding-left:123px;padding-top:4px;" class="control-label col-sm-2" for="email">Source:</label>
                <div class="col-sm-9">
                  <input type="text" class="form-control" id="email" placeholder="Source" name="source">
              </div>
    </div>
  <div class="form-group">
        <label style="padding-left:95px;padding-top:4px" class="control-label col-sm-2" for="email">Destination:</label>
        <div class="col-sm-9">
                <input type="text" class="form-control" id="email" placeholder="Destination" name="destination">
</div>
</div>
        <?php
                if (isset($_POST['passenger_count'])) {
                $count = $_POST['passenger_count'];
                  <h1>$count</h1>

                for ($i=0; $i < $count; $i++) {
         ?>
              <h6><?php echo "Passenger "; echo $i+1;?></h6>
                <div class="form-group">
        <label class="control-label col-sm-2" for="email">Name:</label>
        <div class="col-sm-9">
          <input type="text" class="form-control" id="email" placeholder="Name" name="name<?php echo "$i" ?>">
</div>
</div>
<div class="form-group">
        <label class="control-label col-sm-2" for="email">Age:</label>
        <div class="col-sm-9">
                <input type="text" class="form-control" id="email" placeholder="Age" name="age<?php echo "$i" ?>">
        </div>
 </div>
<?php
        }
  ?>
     <?php
    }
  ?>

        <button onclick="myBooked()" class="btn btn-primary" name="book" style="margin-left: 41%; margin-top: 15px;margin-bottom:10px;">Book Tickets</button>

</form>
```

Ticket Cancelation

```
<?php
        if (isset($_GET['orderid'])) {
                $orderid_cancel = $_GET['orderid'];
                $bus_id = $_GET['bus_id'];

                $query = "DELETE FROM orders WHERE order_id=$orderid_cancel";

                $cancel_order = mysqli_query($connection,$query);

                if (!$cancel_order) {
                        die("Query Failed".mysqli_error($connection));
                }
        }

      $query = "SELECT available_seats FROM posts WHERE post_id=$bus_id";
      $get_seats = mysqli_query($connection,$query);

      while ($row = mysqli_fetch_assoc($get_seats)) {
         $available_seats = $row['available_seats'];
      }

      $query = "UPDATE posts SET available_seats=$available_seats-1 WHERE
post_id=$bus_id";
      $update_seats = mysqli_query($connection,$query);

   ?>
```

Testing

Unit Testing:-

This is the testing process which we can do manually because in this testing program is a tested individually using dummy record to see whether that program produce satisfied output as the company and validation also.

Validation Testing:-

In this requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final Assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple

definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

- Validation test criteria
- Configuration review
- Alpha and Beta testing(conducted by end user)

System Testing:-

System testing is actually a series different test whose primary purpose is to full exercise the computer base system, where the software and other system elements are tested as whole. To test computer software, we spiral out along streamlines that broadens the scope of testing with each turn.

The last higher-order testing step falls outside the boundary of software Engineering and in to the broader context of computer system engineering. Software, once validated it must be combined with other system Elements (e.g. hardware, people, databases). System testing verifies that all the elements Mesh properly and that overall system function/performance is achieved.

1. Recovery Testing
2. Security Testing
3. Stress Testing

Test Cases

Admin Test Cases					
Case ID	Test case	Expected Output	Actual Output	Pass/Fail	Remarks
01	Login Test	Login with proper details and role.	Successfully	Pass	Good
02	Managing details	Manage information properly stored	successfully	Pass	Good
03	Add &update Bus information	Add &update Bus info properly	successfully	pass	Good
04	Exception Handling	Proper error message should display.	Not Successfully	Fail	Poor
05	View tickets booking information	View booking info properly	Successfully	pass	Good
06	Modify User's information	Modifying User Details	successfully	pass	Good

Customer Test cases

Register Customer Test Cases:					
Case ID	Test case	Expected Output	Actual Output	Pass/Fail	Remarks
01	Login Test	Login with proper details	Successfully	Pass	Good
02	Register Information	Proper Register with exception handling	Successfully	Pass	Good
03	View bus Information	Proper Bus Information	Successfully	Pass	Good
04	search information	Search information Properly	Successfully	Pass	Good
05	Book Ticket Information	Book bus tickets Properly	Successfully	Pass	Good
06	Cancel Ticket	Cancel Booked Ticket	Successfully	Pass	Good

Conclusion and Post Implementation Review

After the system is implemented and conversion is completed a review of system is usually conducted by user and analyst. This is called post implementation review.

The most fundamental concern post implementation review to determining whether the system has met its objective; that is analysts want to know if the performance the performance level of improved and if the system is producing the result intended. If neither is happening, one may question whether the system can be considered successful or Not.

The new system needs less manpower, provide facility for Book Tickets and Latest News for bus information. Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

Future Scope for the Project

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printer in future.
- We can give more advance software for Bus Ticket Booking System including more facilities
- We will host the platform on online servers to make it accessible worldwide
- We can add a GPS Tracker to buses so that a customer can know the exact location of the bus.
- Integrate multiple load balancers to distribute the loads of the system
- Create the master and slave database structure to reduce the overload of the database queries
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

References and Bibliography:

- https://www.google.com/ for problem solving
- https://www.apachefriends.org/
- https://www.w3schools.com/php/
- https://www.w3schools.com/html/
- https://www.w3schools.com/php/
- https://www.w3schools.com/css/
- https://www.w3schools.com/php/
- https://www.w3schools.com/javascript/
- https://tutorialspoint.com/php/
- https://www.w3schools.com/php/