* 1. **Database Design**

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. The main objectives of databases are data integrity and data independence.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users

Database Management System (DBMS) allow the data to be protected and organized separately from other resources like hardware, software, and programs. DBMS is a software package, which contains components that are not found other data management packages. The significant of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. That is the difference between the logical and physical data.

The main objectives covered in database design are:

* Controlled redundancy
* Data independence
* Accuracy and integrity
* Privacy and security
* Performance.

The database tables used in this project are given below.

4.**4 Database Design**

### 4.5.1 TABLE DESIGN Table 4.4.1: user\_login

Field name Data type size Constraints Description

id int 11 Primary key id

uid varchar 75 Not null user id

pwd varchar 25 Not null password

type varchar 10 Not null type

st int 2 Not null status

Table 4.4.2 bus\_assign

Field name Data type size Constraints Description

id int 11 Primary key id

bus\_id int 11 Foreign key bus id

station\_id int 11 Foreign key station id

st int 2 Not null status

Table 4.4.3 bus\_data

Field name Data type size Constraints Description

bus\_id int 11 Primary key bus id

bus\_no varchar 25 Not null bus number

no\_plate varchar 25 Not null number plate

st int 3 Not null status

Table 4.4.4 daily\_trip

Field name Data type size Constraints Description

Id int 11 Primary key id

bus\_id int 11 Foreign key bus id

date date Not null date

conductor varchar 150 Not null conductor

trip\_id int 11 Foreign key trip id

path\_id int 11 Foreign key path id

amt int 11 Not null amount

st int 3 Not null status

Table 4.4.5 district

Field name Data type size Constraints Description

dist\_code int 11 Primary key district code

st\_code int 11 Not null bus number

Distname varchar 200 Not null distrct name

Table 4.4.6 path\_data

Field name Data type size Constraints Description

p\_id int 11 Primary key path id

route\_id int 11 Foreign key route id

path\_name varchar 150 Not null path name

st int 3 Not null status

Table 4.4.7 fare\_data

Field name Data type size Constraints Description

id int 11 Primary key id

p\_id int 11 Foreign key path id

frm\_jnname varchar 100 Not null from junction

to\_jnname varchar 100 Not null from junction

frm\_jnid int 11 Not null from junction id

to\_jnid int 11 Not null to junction id

amt int 11 Not null amount

st int 3 Not null status

Table 4.4.8 path\_info

Field name Data type size Constraints Description

id int 11 Primary key id

p\_id int 11 Foreign key path id

dis int 11 Not null route id

jn\_name varchar 150 Not null junction name

st int 3 Not null status

Table 4.4.9 state

Field name Data type size Constraints Description

Stcode int 11 Primary key state code

statename varchar 150 Not null state name

Table 4.4.10 route\_data

Field name Data type size Constraints Description

rt\_id int 11 Primary key state code

source\_id int 11 Not null source id

source\_code varchar 25 Not null source code

dest\_id int 11 Not null destination id

dest\_code varchar 25 Not null destination code

st int 3 Not null status

route name varchar 150 Not null route name

Table 4.4.11 trip\_data

Field name Data type size Constraints Description

trip\_id int 11 Primary key trip id

bus\_id int 11 Foreign key bus id

station\_id int 11 Foreign key station id

start\_time varchar 25 Not null start time

st int 3 Not null status

trip\_nme varchar 50 Not null trip name

Table 4.4.12 staff\_assign

Field name Data type size Constraints Description

as\_id int 11 Primary key assign id

bus\_id int 11 Foreign key bus id

stf\_id int 11 Foreign key staff id

st int 3 Not null status

type varchar 50 Not null type of staff

Table 4.4.13 staff\_data

Field name Data type size Constraints Description

st\_id int 11 Primary key staff id

nme varchar 150 Not null name

addr text Not null address

con varchar 12 Not null contact

dob date Not null date of birth

uid varchar 25 Not null user id

pic varchar 25 Not null picture

st int 3 Not null status

styp int 11 Not null type of staff

Table 4.4.14 station\_data

Field name Data type size Constraints Description

id int 11 Primary key station id

state varchar 75 Not null state

district varchar 75 Not null district

loc varchar 25 Not null location

is\_st int 11 Not null is stand

st\_code varchar 10 Not null state code

pic varchar 25 Not null picture

uid varchar 50 Foreign key user id

addr text Not null address

con varchar 25 Not null contact

st int 3 Not null status

em varchar 125 Not null email

Table 4.4.15 trip\_info

Field name Data type size Constraints Description

id int 11 Primary key id

trip\_id int 11 Foreign key trip id

from\_id int 11 Foreign key from id

station\_id int 11 Foreign key station id

route\_id int 11 Foreign key route id

p\_id int 11 Foreign key path id

st int 3 Not null status

Table 4.4.16 user\_data

Field name Data type size Constraints Description

id int 11 Primary key id

adrnum varchar 30 Not null adhar number

accnum varchar 20 Not null account number

nme varchar 20 Not null name

con varchar 12 Not null contact

pic varchar 30 Not null picture

dob date Not null date of birth

st int 3 Not null status

Table 4.4.17 ticket\_issue

Field name Data type size Constraints Description

id int 11 Primary key id

dt date Not null date

daily\_trip\_id int 11 Foreign key daily trip id

qr\_id varchar 50 Not null qr id

srn varchar 50 Not null source name

srid int 11 Not null source id

dsn varchar 50 Not null destination name

did int 11 Not null destination id

amt int 11 Not null amount

st int 3 Not null status

Table 4.4.18 dailt\_trip\_stop

Field name Data type size Constraints Description

id int 11 Primary key id

dt date Not null date

daily\_tripid int 11 Not null daily trip id

jn\_name varchar 25 Not null junction name

jn\_id int 11 Not null junction id

time varchar 10 Not null time

Table 4.4.19 vb\_acno

Field name Data type size Constraints Description

id int 11 Primary key id

num int 11 Not null number

Table 4.4.20 vd\_cdepo

Field name Data type size Constraints Description

id int 11 Primary key id

amt\_frm varchar 250 Not null amount from

amt\_to varchar 250 Not null amount to

amt big int 20 Not null amount

dt date Not null date

desc text Not null descripton

st int 11 Not null status

Table 4.4.21 vb\_usr

Field name Data type size Constraints Description

id int 11 Primary key id

nme varchar 250 Not null name

em varchar 250 Not null email

con bigint 20 Not null contact

addr varchar 250 Not null address

acc\_num varchar 250 Not null account number

acc\_amt big int 20 Not null account amount

pwd varchar 250 Not null password

**CHAPTER 5 SYSTEM TESTING**

* 1. **5.1 Introduction**

In any software development, testing is a process to show the correctness of the program and it meets the design specifications. Testing is needed to prove correctness, to show completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary to ensure completeness of testing, improve the quality of the software, and reduce the testing costs and to reduce study needs and operation time. Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. A series of test cases are created that are intended to demolish the software that has been built.

Based on the way the software reacts to these tests, we can decide whether the configuration that has been built is study or not. It is essential that all components of an application be tested, as the failure to do so many result in a series of bugs after the software is put to use. Several methods of testing exist in software Engineering, which enable a programmer to make sure that the configuration built is free of bugs.

### 5.2 Testing Process

For any software that is newly developed, first and foremost preference is given to the testing of the system. It is developer‘s last chance to detect and correct the errors. That may occur possibly in the software. The programmers will generate a set of test data, which will give the maximum possibility of finding all most all types of errors that can occur in the system.

### 5.2.1 Unit Testing

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between the modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

The most common approach to unit testing requires drivers and stubs to be written. The driver simulates a calling unit and the stub simulates a called unit. The investment of developer time in this activity sometimes results in demoting unit testing to a lower level of priority and that is almost always a mistake. Even though the drivers and stubs cost time and money, unit testing provides some undeniable advantages. It allows for automation of the testing process, reduces difficulties of discovering errors contained in more complex pieces of the application, and test coverage is often enhanced because attention is given to each unit.

### 5.2.2 Integration Testing

Integration testing (sometimes called Integration and Testing, abbreviated as ―I&T‖) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes place as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These ―design items‖ i .e. assemblages (or groups of units) are exercised through their interfaces using Black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing. The overall idea is a ―building block‖ approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

### 5.2.3 Validation Testing

Data validation is the process of testing the accuracy of data. A set of rule we can apply to a control to specify the type and range of data that can enter. It can be used to display error alert when users enter incorrect values in to a form.

Now performing validation testing in system Centralized Social Welfare by undergoing validation for each tools and the validation succeeded when the software function in a manner that can be reasonably accepted, by the user.

### 5.2.4 User Acceptance Testing

User acceptance of a system is a key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly, keeping in touch with the prospective system user at the time of developing and making changes whenever required.

### 5.5 Testing strategies

* + 1. **5.5.1 Top-Down Testing**

Top-Down Testing tests the higher levels of a system before testing its detailed components. The program is represented as a single abstract component with sub components represented by stubs. Stubs have the same interface as the components but very limits functionally. After the top-level component has been tested, its sub-components are implemented and tested in the same way. This process continues recursively until the bottom- level components are implemented. The whole system may then be completely tested.

### 5.5.2. Bottom-Up Testing

Bottom-Up Testing is the converse of Top-Down Testing. It involves testing the modules at the lower levels of the hierarchy and then working up the hierarchy of the modules until the final module is tested. The advantage of bottom-up testing is the disadvantage of top- down testing and vice versa. When using bottom-up testing test drivers must be written to exercise the lower level components. These test drivers simulate the components environment and are valuable components; the test drivers and test data should be distributed with the component. Potential re-users can then run these tests to satisfy themselves that the component behaves as expected in their environment.

### 5.5.3.Black Box testing

Knowing the specified function that a product has been designed to perform, test can be conducted that demonstrates each function that is fully operational, at the same time searching for errors in each function. Black Box testing focuses on functional requirement of the software. Black Box testing attempts to find out errors in the following categories.

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access.
* Performance and errors
* Initialization and termination errors

### 5.5.4. White Box Testing

Knowing the internal working of a product test can be conducted to ensure that ―all gears mesh‖ that is internal operation performs according to specification and all internal components have been adequately exercised. Using white box testing methods, the software engineer can derive test cases that

* Guarantee that all independent paths within a module have been exercised at least once basis path testing.
* Exercise all logical decisions on their true and false sides- Condition testing.
* Execute all loops at their boundaries and within their operation bounds- Loop testing.
* Exercise internal data structures to assure their validity-data flow testing.

### 5.6 TEST CASES

Test cases are the key to the process because they identify and communicate the conditions that will be implemented in test and are necessary to verify successful and acceptable implementation of the product requirement. They are all about making sure that the product fulfils the requirements of the system.

Test case 1

1. Project Title : Smart journey
2. Software Tool : PHP

3. Test objective :To check whether the entered User name and Password

are valid or invalid.

4. Test data : Username = admin and password = admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StStep no:** | **StSteps** | **D Data** | **E Expected results** | **AcActual results** |
| 11  1 | EnEnter User name and press LOGIN Button | usUsername=admin | ShShould display warning message box "Incorrect username or password" | LoLogin failed |
| 2  2 | EnEnter Password and press LOGIN Button | paPassword=admin | ShShould display warning message box " Incorrect username or password " | LoLogin failed |
| 323 | Enter User name and Password and press LOGIN Button | username=admin and password=123 | ShShould display warning message box " Incorrect username or password" | LoLogin failed |
| 444 | EnEnter User name and Password and press LOGIN Button | usUsername=abc and password= admin | ShShould display warning message box " Incorrect username or password" | LoLogin failed |
| 555 | EnEnter User name and Password and press LOGIN Button | usUsername= abc and password = 123 | ShShould display warning message box " Incorrect username or password" | LoLogin failed |
| 666 | EnEnter User name and Password and press LOGIN Button | usUsername=admin and password=admin | ShShould navigate to admin home page | LoLogin success |

### CHACHAPTER 6

### SYSTEM IMPLEMENTATION

**6.1 Introduction**

The final and important phase in the system life cycle is the implementation of the new system. The term implementation has different meanings ranging from the conversion of a basic application to a complete replacement of a computer system. The procedure however, is virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing existing system, manual or automated, or it may be a major modification to an existing system.

The method of implementation and time scale to be adopted is found out initially. Next the system is tested properly and at the same time users are trained in the new procedure. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but it will prevent improper installation.

The implementation involves following things:

* Careful planning.
* Investigation of the system considerations.
* Design the method to achieve the changeover.
* Evaluation of change over method.

Implementation of a new system requires the operating staff installing the software and creating computer files. There are many ways in which this can be achieved. The most common methods are the following.

* Direct change over
* Parallel running
* Pilot running change over The creation of the designed system takes place in the implementation phase.

This phase activities do the following:

* Development of phase overview
* Preparing for implementation
* Computer program development
* Development phase report and overview

It also performs activities like writing, testing, debugging and documenting the programs. There are three types of implementations:

* Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.
* Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned, there can be many problems. Some large computer systems have taken as long as a year to convert.
* Implementation of a modified application to replace the existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files. Every system requires periodic evaluation after implementation.

This is to review the performance of the system and to evaluate against established standard or criteria. A study is conducted for measuring the performance of the system against pre- defined requirements. This study results a post-implementation review that determines how well the system continues to meet the performance specification.

### CHAPTER 7 CONCLUSION

This software project is meant for the analysis and planning of the activities of a passenger transport system.The objective of the application is to make the system efficient and profitable.The existing system operates different schedules from various depots to multiple destinations either as short or long trips. As the numbers of schedules are more,it is difficult to get daily report of profit or loss.The proposed system facilitates to provide such information by text or graphical charts.Prepaid travel cards can be introduced for passengers which can be recharged online or offline.Traveller can swipe the travel card in the bus for his journey.The amount will be deducted in the card as well as the central server accordingly. Parallel the conventional ticketing system also should be operational,as all the commuters cannot avail such advanced cards.This data will also be updated at the end of the day.The system can generate different types of reports based on daily data for the analysis and planning.The information can be accessible even to the top management for policy making.The analysis and planning module of the system suggest solutions for making the service profitable such as re-routing,re-scheduling or even partial cancellation of schedules. This generic system which can be applied to any transport service system.

### CHAPTER 8

### FUTURE ENHANCEMENT

Today the system is focusing on the development of different welfare organisations. At the beginning stage the website is free for all the users. The organisations have the facility to feature their exhibited products .In future the organisations also should pay some amount of money to the admin for adding products for exhibitions.

**APPENDIX B-SAMPLE CODE**

**connection.php**

<?php

mysql\_connect("localhost","root","");

mysql\_select\_db("smart\_journey");

?>

**Login.php**

﻿<?php

ob\_start();

include './connection.php';

if(isset($\_POST['sub']))

{

$un=$\_POST['un'];

$pas=$\_POST['pas'];

$chk\_login=mysql\_query("select \* from user\_login where uid='$un' and pwd='$pas'");

if(mysql\_num\_rows($chk\_login)>0)

{

session\_start();

$r\_login=mysql\_fetch\_row($chk\_login);

if($r\_login[3]=="admin")

{

$\_SESSION['adm']=$un;

header("location:Admin/index.php");

}

if($r\_login[3]=="stand")

{

$\_SESSION['stnd']=$un;

header("location:Stand/index.php");

}

if($r\_login[3]=="conductor")

{

$\_SESSION['con']=$un;

header("location:Conductor\\index.php");

}

if($r\_login[3]=="checker")

{

$\_SESSION['con']=$un;

header("location:Checker//index.php");

}

}

else

{

header("location:login.php?error=1");

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">

<title>Intelligent Business Analyser</title>

<!-- Favicon-->

<link rel="icon" href="\_Template/favicon.ico" type="image/x-icon">

<!-- Google Fonts -->

<link href="https://fonts.googleapis.com/css?family=Roboto:400,700&subset=latin,cyrillic-ext" rel="stylesheet" type="text/css">

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" type="text/css">

<!-- Bootstrap Core Css -->

<link href="\_Template/plugins/bootstrap/css/bootstrap.css" rel="stylesheet">

<!-- Waves Effect Css -->

<link href="\_Template/plugins/node-waves/waves.css" rel="stylesheet" />

<!-- Animation Css -->

<link href="\_Template/plugins/animate-css/animate.css" rel="stylesheet" />

<!-- Custom Css -->

<link href="\_Template/css/style.css" rel="stylesheet">

</head>

<body class="login-page">

<div class="login-box">

<div class="logo">

<a href="javascript:void(0);">Business<b>Analyser</b></a>

<small>for Transport Department</small>

</div>

<div class="card">

<div class="body">

<form id="sign\_in" method="POST">

<div class="msg">Sign in to start your session</div>

<div class="input-group">

<span class="input-group-addon">

<i class="material-icons">person</i>

</span>

<div class="form-line">

<input type="text" class="form-control" name="un" placeholder="Username" required autofocus>

</div>

</div>

<div class="input-group">

<span class="input-group-addon">

<i class="material-icons">lock</i>

</span>

<div class="form-line">

<input type="password" class="form-control" name="pas" placeholder="Password" required>

</div>

</div>

<div class="row">

<div class="col-xs-8 p-t-5">

<input type="checkbox" name="rememberme" id="rememberme" class="filled-in chk-col-pink">

<label for="rememberme">Remember Me</label>

</div>

<div class="col-xs-4">

<input type="submit" name="sub" value="Login here" class="btn btn-success" />

</div>

**CHAPTER 2**

**SYSTEM ANALYSIS**

System analysis is the starting point for system design. System Analysis is a detailed study of various operations performed by a system and its relationships within and outside the system. The basic aim of system analysis is to obtain a clear understanding of the needs of all users, what exactly is decided from the software and what the constraints on the solutions. The analysis of the role of proposed system and the identification of the requirements that it should meet the strength and weakness is determined for the better design of the system. Once the analysis is complete, the analyst should have a firm understanding of what is to requirements can be implemented before system

</div>

<center>

<?php

if(isset($\_GET['error']))

{

if($\_GET['error']=="1")

{

echo"<font color='red'>Invalid User ID or Password</font>";

}

}

?>

</center>

</form>

</div>

</div>

</div>

<!-- Jquery Core Js -->

<script src="\_Template/plugins/jquery/jquery.min.js"></script>

<!-- Bootstrap Core Js -->

<script src="\_Template/plugins/bootstrap/js/bootstrap.js"></script>

<!-- Waves Effect Plugin Js -->

<script src="\_Template/plugins/node-waves/waves.js"></script>

<!-- Validation Plugin Js -->

<script src="\_Template/plugins/jquery-validation/jquery.validate.js"></script>

<!-- Custom Js -->

<script src="\_Template/js/admin.js"></script>

<script src="\_Template/js/pages/examples/sign-in.js"></script>

</body>

</html>

**Index.php**

<?php

include './connection.php';

if(isset($\_POST['sub']))

{

$s=$\_POST['station'];

$d=$\_POST['station1'];

//find the junction id

$sel\_p1=mysql\_query("select distinct p\_id from path\_info where jn\_nme='$s' || jn\_nme='$d'");

if(mysql\_num\_rows($sel\_p1)>0)

{

while($r1=mysql\_fetch\_row($sel\_p1))

{

$pathid=$r1[0];

//finding the id from the table to find the bus direction

$sel1=mysql\_query("select id from path\_info where p\_id='$pathid' and jn\_nme='$s'");

$sel2=mysql\_query("select id from path\_info where p\_id='$pathid' and jn\_nme='$d'");

if(mysql\_num\_rows($sel1)>0 && mysql\_num\_rows($sel2)>0)

{

$sid=mysql\_fetch\_row($sel1);

$sourceid=$sid[0];

$did=mysql\_fetch\_row($sel2);

$destid=$did[0];

if($sourceid<$destid)

{

//find active trip

$sel\_t=mysql\_query("select \* from daily\_trip where path\_id='$pathid' and st=0");

$r\_t=mysql\_fetch\_row($sel\_t);

$tripid=$r\_t[0];

$busid=$r\_t[2];

//find current bus position

$sel\_pos=mysql\_query("select \* from dailt\_trip\_stop where daily\_tripid='$tripid' order by id desc");

$r\_pos=mysql\_fetch\_row($sel\_pos);

$current\_stop=$r\_pos[3];

$stopid=$r\_pos[4]; // to find the passanger

//find the stop id of the bus

$sel\_stid=mysql\_query("select \* from path\_info where p\_id='$pathid' and jn\_nme='$current\_stop'");

$r\_stid=mysql\_fetch\_row($sel\_stid);

$curnt\_stopid=$r\_stid[0];

if($curnt\_stopid<$sourceid)

{

$sel\_bus=mysql\_query("select \* from bus\_data where bus\_id='$busid'");

$r\_bus=mysql\_fetch\_row($sel\_bus);

?>

<div class="col-lg-4">

<img src="Admin/bus\_pic/<?php echo $r\_bus[4] ?>" class="img img-responsive" />

Current Position : <?php echo $current\_stop ?><br />

Total Passengers : <?php $sel\_tcb=mysql\_query("select \* from ticket\_issue where daily\_tripid='$tripid' and did>$stopid");

echo mysql\_num\_rows($sel\_tcb); ?>

</div>

<?php

}

}

}

}

}

}

?>

<html>

<head>

<script src="\_Template/plugins/jquery/jquery.min.js"></script>

<link href="\_Template/plugins/bootstrap/css/bootstrap.css" rel="stylesheet">

</head>

<body>

<a href="login.php">LOGIN</a>

<form method="post">

<script>

function loaddis(x)

{

$("#dd").load("getdis1.php?q="+x)

}

function loaddis1(x)

{

$("#dd1").load("getdis2.php?q="+x)

}

function loadst1(x)

{

$("#st").load("getstation1.php?did="+x);

}

function loadst2(x)

{

$("#st1").load("getstation2.php?did="+x);

}

</script>

<table class="table table-hover table-responsive table-bordered table-condensed table-striped">

<tr>

<td>

<select name="stat" class="form-control show-tick" onchange="loaddis(this.value)">

<option value="0">Choose</option>

<?php

$sel\_st=mysql\_query("select \* from state");

while($r\_st=mysql\_fetch\_row($sel\_st))

{

?>

<option value="<?php echo $r\_st[0] ?>"><?php echo $r\_st[1] ?></option>

<?php

}

?>

</select>

</td>

<td>

<select name="stat" class="form-control show-tick" onchange="loaddis1(this.value)">

<option value="0">Choose</option>

<?php

$sel\_st=mysql\_query("select \* from state");

while($r\_st=mysql\_fetch\_row($sel\_st))

{

?>

<option value="<?php echo $r\_st[0] ?>"><?php echo $r\_st[1] ?></option>

<?php

}

?>

</select>

</td>

</tr>

<tr>

<td>

<span id="dd"></span>

</td>

<td>

<span id="dd1"></span>

</td>

</tr>

<tr>

<td>

<span id="st"></span>

</td>

<td>

<span id="st1"></span>

</td>

</tr>

<tr>

<td colspan="2">

<center>

<input type="submit" name="sub" value="ADD ROUTE" class="btn btn-danger" />

</center>

</td>

</tr>

</table>

</form>

</body>

</html>

**Logout.php**

<?php

session\_start();

session\_destroy();

header("location:index.php");

?>

### REFERENCES

**BOOKS REFERRED**

Patrick Naughton Herbert Schildt ….The Complete Reference Java2, Tata Mcgraw….Hill Edition

Avi Silberschatz, Henry F. Korth and S. Sudarshan ----Database system concept Programming in java ----- Balaguruswamy

James Keogsh ----- J2EE: The complete Reference, first Edition

V Rajaraman ------ Analysis and Design of Information Systems, PHI

Roger Pressman, Bruce Maxim ----- Software Engineering: A Practitioner's Approach ----- 8th Edition Jon Duckett ----- JavaScript & jQuery: Interactive Front-End Web Development.

### WEB SITES

<http://www.javatpoint.com/> <http://www.tutorialspoint.com/java/> <http://www.w3schools.com/js/js>syntax.asp <http://www.roseindia.net/jsp/index3.shtml>