1. INTRODUCTION

Tourism Management System is a complete tourist fully integrated tourism web site. The website covers all the areas required for an including tourism. This project is developed to manage the tourist in the tourism management web site. The main module in this project is login, tourist management, complaints and reports.

The objective of the project is to develop a system that automates the processes and activities of a travel and. The purpose is to design a system using which one can perform all operations related to traveling.

1.1. Organization Profile

Over 7 years old company, RenewBuy was started by Indraneel Chatterjee and Balachander Sekhar. RenewBuy is a fully technology-integrated, industry-focused insurance and financial products consulting firm. Our highly trained and specialized advisors are peer-level consultants who serve as an extension of our customer's financial and insurance needs to proactively elevate customers' experience, simplify processes, and make them more accessible. RenewBuy is a proud people-driven organization with a reputation for technology innovation, service, and culture.

With a strong network of over 90,000 advisors spread across 750+ cities, RenewBuy envisions providing financial and insurance products across the length and breadth of the country. RenewBuy is branched across 60+ offices with a total employee strength of 2000+ and works with over 35+ insurers across motor, health, and life insurance categories and financial products in loans, credit cards, and health plans.

Our Core idea, vision, and mission

Vision: Insurance and Financial products simplified and made accessible to all

Mission: Deliver the widest choice of insurance and financial products on a reliable platform through digitally enabled advisors

Values: Execution excellence - We will deliver high-quality & innovative thinking to provide reliable solutions

1.2 System Specifications

1.2.1 HARDWARE CONFIGURATION

Processor	: Pentium -IV
Speed	: 1 GHz
Hard Disk Capacity	: 40GB
RAM Capacity	: 1GB RAM
CD-ROM Drive	: 52x speed
Keyboard	: 104 keys
Mouse	: Logitech
Printer	: HP3745 series DeskJet printer

1.2.2. SOFTWARE SPECIFICATION

Operating System	:	Windows 7/8/10
Front End	:	JAVA
Back End	:	MYSQL

2. SYSTEM STUDY

Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility of a proposed solution is evaluated in teams of its components. These components are:

- Economic feasibility
- Technical feasibility

Economic Feasibility

The economic feasibility study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development or the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

The technical feasibility study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The will lead to high demands on the available technical resources. This will lead to high demands being places on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

2.1. EXISTING SYSTEM

In the present system a customer has to approach various agencies to find details of places and book tickets. This often requires a lot of time and effort. A customer may not get desired information from these offices and often the customer may be misguided. It is tedious for a customer to plan a particular journey and have it executed properly.

2.1.1. DRAWBACKS

- This system is not an user friendly
- Can't use all users

2.2. PROPOSED SYSTEM

The proposed system is a web-based application and maintains centralized repository of all related information. The system allows one to easily access the relevant information and make necessary arrangements. User can decide about places they want to visit and make bookings online for travel and accommodation

2.2.1. FEATURES

- User no needs to google it.
- Amazing web site design

3. SYSTEM DESIGN AND DEVELOPMENT

Design is concerned with identifying software components specifying relationship Among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is Minimal clearly specified. Design will explain software components in details. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the further requirements.

The design document describes how to transform, the requirement and the functional design into more technical system design specification. This design involves conceiving and planning out in the mind and making a drawing pattern of sketch of. It includes type of activities, External Design, Architectural Design and Detailed Design. The architectural design and detailed design collectively referred to as internal design.

The external design involves specifying the externally observable characteristics of a software product and the internal design involves specifying the internal structure and processing details of the system. The fundamental concept of the design includes abstraction structure, information hiding Modularity, concurrency, verification and design aesthetics.

3.1 FILE DESIGN

In computing, a file design (or file system) is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into individual pieces, and giving each piece a name, the information is easily separated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names are called a "file system".

Some file systems are used on local data storage devices; others provide file access via a network protocol. Some file systems are "virtual", in that the "files" supplied are computed on request or are merely a mapping into a different file system used as a backing store. The file system manages access to both the content of files and the metadata about those files. It is responsible for arranging storage space; reliability, efficiency, and tuning with regard to the physical storage medium are important design considerations.

3.2 INPUT DESIGN

The input design is the process of entering data to the system. The input design goal is to enter to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operations are minimized.

The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using it.

Input design is the process of converting user-originated inputs to a computer-based format input data are collected and organized into group of similar data. Once identified, appropriate input media are selected for processing.

The input design also determines the user to interact efficiently with the system. Input design is a part of overall system design that requires special attention because it is the common source for data processing error. The goal of designing input data is to make entry easy and free from errors.

Input design is the process of connecting the user-originated inputs into a computer to used format. The goal of the input design is to make the data entry logical & free from errors.

3.3 OUTPUT DESIGN

Output design is the process of converting computer data into hard copy that is understood by all. The various outputs have been designed in such a way that they represent the same format that the office and management used to.

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the systems relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the Data Flow Diagram (DFD). Human factors educe issues for design involves addressing internal controls to ensure readability.

The output form in the system is either by screen or by hard copies. Output design aims at communicating the results of the processing of the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels.

All reports are output formats, maintained details can be reported over crystal reports, this project sustain following reports

3.4 DATABASE DESIGN

The most important consideration in designing the database is how information will be used.

The main objectives of designing a database are:

Data Integration

In a database, information from several files are coordinated, accessed and operated upon as through it is in a single file. Logically, the information are centralized, physically, the data may be located on different devices, connected through data communication facilities.

Data Integrity

Data integrity means storing all data in one place only and how each application to access it. This approach results in more consistent information, one update being sufficient to achieve a new record status for all applications, which use it. This leads to less data redundancy; data items need not be duplicated; a reduction in the direct access storage requirement.

Data Independence

Data independence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of applications and to allow modifications to application programs without reorganizing the physical data.

The tables needed for each module were designed and the specification of each and every column was given based on the records and details collected during record specification of the system study.

3.5 SYSTEM DEVELOPMENT

The key to control maintenance costs is to design systems that are easy to change, so the link between development and maintenance is very strong. Many of the analysis and design methodologies, tools, and techniques employed during system development can be applied to system maintenance, but there are significant differences between development and maintenance. Maintainability is the ease with which software can be understood, corrected, adopted and enhanced.

3.5.1 DESCRIPTION OF MODULES

To develop this project several step should be followed. There are various modules in this proposed system they are listed below.

- Create User Account
- Create & Manage Tour Package
- Booking Places
- Manage Booking
- Manage Enquiries

Create User Account:

User can manually open this application and create the account, once the account has been created then user can able to login by using username and password. After the login only user can select the package and booking their favorite places.

Create & Manage Tour Package:

Admin can create and modify the package details, this screen has to collect some information for package name, package type, location, price, images etc. This packages are displaying in the user account and they can book the package.

Book Places:

The package details are displaying in the user portal then the user can check the package details then they can check and book the places. Once the packages are booked then admin will get notify.

Manage Booking:

Once the packages are booked then the admin will get notify and view the booking details. The details have user details as well then, they can contact them through mobile. It s an main module for admin can contact them.

Managing Enquiries:

If the user wants to collect the enquiries, they have to submit the enquiry details; it will be shown in the admin portal. If the admin wants to clarify their enquiry, they will contact them.

4. TESTING AND IMPLEMENTATION

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

Integration Testing

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

Validation Testing

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

Output Testing

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

IMPLEMENTATION

System implementation is the stage of the project that the theoretical design is turned into a working system. If the implementation stage is not properly planned and controlled, it can cause error. Thus it can be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Normally this stage involves setting up a coordinating committee, which will act as a sounding board for ideas; complaints and problem. The first task is implementation planning; i.e., deciding on the methods and time scale to be adopted. Apart from planning two major task of preparing for implementation are, education takes place much earlier in the project; at the implementation stage the emphasis must be on training in new skills to give staff confidence they can use the system. Once staff has been trained, the system can be tested.

After the implementation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance is to bring the new system to standards.

5. CONCLUSION

In conclusion, a tourist guide management system can offer several advantages to tour companies and guides. These systems help streamline the management of tour guides and clients, improve communication, and increase efficiency in booking and scheduling tours.

Tourist guide management systems can save time and money, reduce errors, and enhance customer satisfaction by improving the overall efficiency of the tour company. By automating routine tasks such as tour scheduling and customer data management, tour company staff can focus more on providing quality service to clients.

However, like any technology, tourist guide management systems do have some drawbacks, such as initial costs, maintenance requirements, and staff training. Tour companies need to consider these factors when deciding whether to implement a management system in their operations.

Overall, a tourist guide management system is a valuable tool for any tour company looking to improve its operations and customer service. By choosing the right system and investing in staff training, tour companies can reap the benefits of automation while providing high-quality service to their clients..

BIBLIOGRAPY

Books Referred:

- Shildt, Herbert. Java: A Beginner's Guide. McGraw-Hill Education, 2018.
- Sierra, Kathy, and Bert Bates. Head First Java, 2nd Edition. O'Reilly Media, 2005.
- Subramaniam, Venkat. Functional Programming in Java: Harnessing the Power of Java 8 Lambda Expressions. Pragmatic Bookshelf, 2014.
- Walrath, Kathy, et al. The Java Tutorial: A Short Course on the Basics. Addison-Wesley Professional, 2018.

Websites:

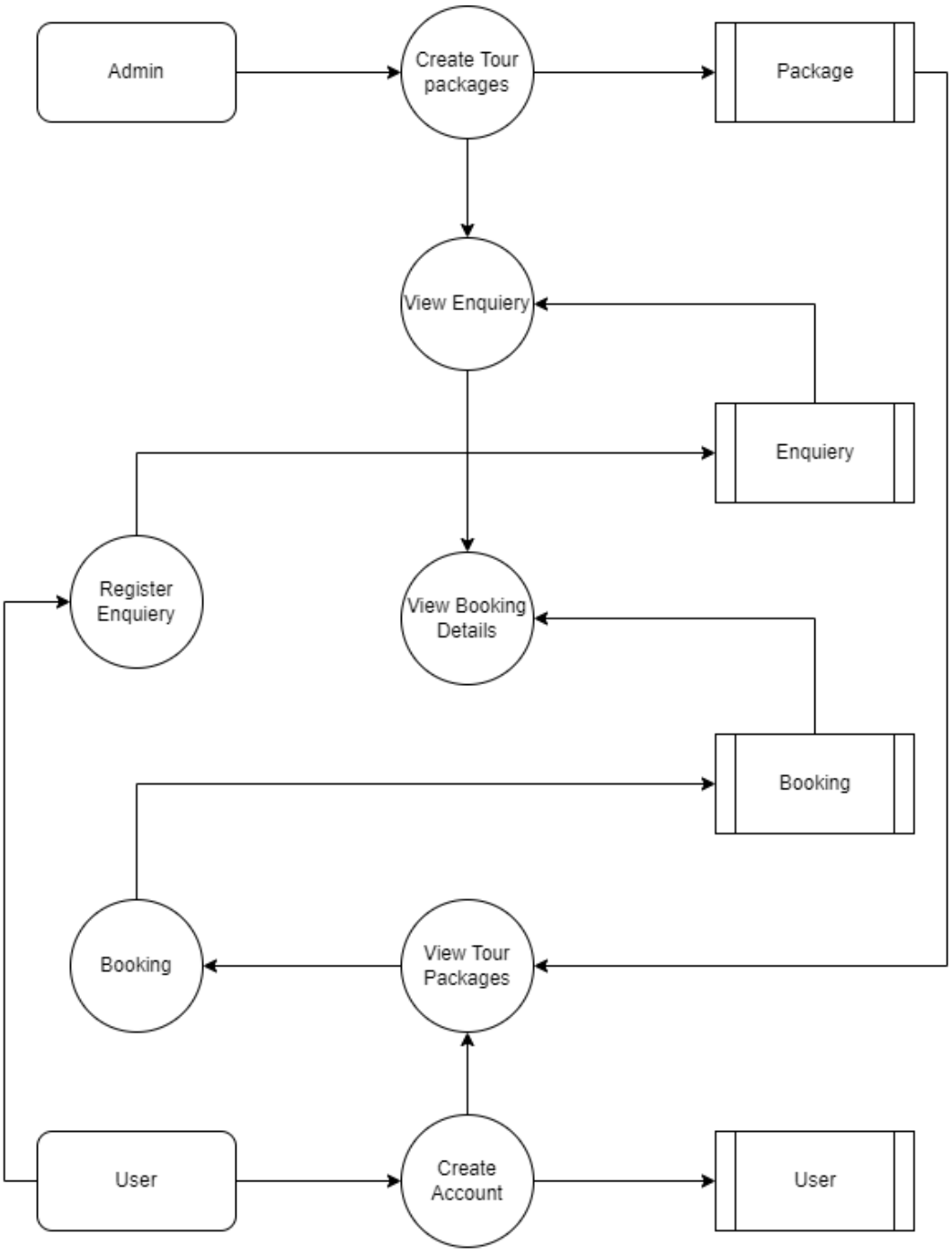
- GeeksforGeeks. "Java Programming Language." GeeksforGeeks, 2023, https://www.geeksforgeeks.org/java/.
- Stack Overflow. "Questions tagged [java]." Stack Overflow, https://stackoverflow.com/questions/tagged/java.
- Tutorials Point. "Java Tutorial." Tutorials Point, 2023, https://www.tutorialspoint.com/java/index.htm.

APPENDICES

A. Data Flow Diagram

Level 0

Level 1


Admin
Create Tour packages
Package
View Enquiery
Enquiery
Register Enquiery
View Booking Details
Booking
Booking
View Tour Packages
User
Create Account
User

B. TABLE STRUCTURE

Table Name **:** Admin

Primary Key **:** Admin_id

Table Description **:** This table is used to maintain the details about Admin

FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
Admin_id	Int	8	Primary Key
Username	Varchar	15	Not null
Password	Varchar	15	Not null

Table Name **:** Package

Primary Key **:** Package_id

Table Description **:** This table is used to maintain the details about Package

FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
Package_id	Int	8	Primary Key
Package name	Varchar	15	Not null
Package type	Varchar	10	Not null
Location	Varchar	20	Not null
Price	Int	10	Not null
Details	Varchar	30	Not null

Table Name **:** Booking

Primary Key **:** Booking_id

Table Description **:** This table is used to maintain the details about Booking

FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
Booking_id	Int	8	Primary Key
Package id	Int	10	Foreign key
User email	Varchar	20	Not null
From date	Date	10	Not null
To date	Date	10	Not null
Comment	Varchar	30	Not null
Reg Date	Date	10	Not null

Table Name **:** User

Primary Key **:** User_id

Table Description **:** This table is used to maintain the details about User

FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
User_id	Int	8	Primary Key
Full Name	Varchar	25	Not null
Mobile	Int	10	Not null
Email	Varchar	15	Not null
Password	Varchar	15	Not null
Gender	Varchar	10	Not null

B. Sample Coding

```
. <?java
session_start();
include('includes/config.java');
if(strlen($_SESSION['alogin'])==0)
        {
header('location:index.java');
}
else{
?>
<!DOCTYPE HTML>
<html>
<head>
<title>TMS | Admin Dashboard</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); },
false); function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- Bootstrap Core CSS -->
<link href="css/bootstrap.min.css" rel='stylesheet' type='text/css' />
<!-- Custom CSS -->
<link href="css/style.css" rel='stylesheet' type='text/css' />
<link rel="stylesheet" href="css/morris.css" type="text/css"/>
<!-- Graph CSS -->
<link href="css/font-awesome.css" rel="stylesheet">
<!-- jQuery -->
<script src="js/jquery-2.1.4.min.js"></script>
<!-- //jQuery -->
<link href='//fonts.googleapis.com/css?family=Roboto:700,500,300,100italic,100,400' rel='stylesheet'
type='text/css'/>
<link href='//fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>
```

```
<!-- lined-icons -->
<link rel="stylesheet" href="css/icon-font.min.css" type='text/css' />
<!-- //lined-icons -->
</head>
<body>
  <div class="page-container">
  <!--/content-inner-->
<div class="left-content">
        <div class="mother-grid-inner">
<!--header start here-->
<?java include('includes/header.java');?>
<!--header end here-->
            <ol class="breadcrumb">
          <li class="breadcrumb-item"><a href="index.html">Home</a> <i class="fa fa-angle-right"></i></li>
        </ol>
<!--four-grids here-->
            <div class="four-grids">
                    <div class="col-md-3 four-grid">
                        <div class="four-agileits">
                            <div class="icon">
                                <i class="glyphicon glyphicon-user" aria-
hidden="true"></i>
                            </div>
                            <div class="four-text">
                                <h3>User</h3>
                                <?java $sql = "SELECT id from tblusers";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=$query->rowCount();
                    ?>                  <h4> <?java echo htmlentities($cnt);?> </h4>
                            </div>
                        </div>
                    </div>
```

```
                                        <div class="col-md-3 four-grid">
                                                <div class="four-agileinfo">
                                                        <div class="icon">
                                                                <i class="glyphicon glyphicon-list-alt" aria-
hidden="true"></i>
                                                        </div>
                                                        <div class="four-text">
                                                                <h3>Bookings</h3>
                                                                        <?java $sql1 = "SELECT
BookingId from tblbooking";
$query1 = $dbh -> prepare($sql1);
$query1->execute();
$results1=$query1->fetchAll(PDO::FETCH_OBJ);
$cnt1=$query1->rowCount();
                                        ?>
                                                                <h4><?java echo htmlentities($cnt1);?></h4>
                                                        </div>
                                                </div>
                                        </div>
                                        <div class="col-md-3 four-grid">
                                                <div class="four-w3ls">
                                                        <div class="icon">
                                                                <i class="glyphicon glyphicon-folder-open" aria-
hidden="true"></i>
                                                        </div>
                                                        <div class="four-text">
                                                                <h3>Enquiries</h3>
                                                                                <?java $sql2 =
"SELECT id from tblenquiry";
$query2= $dbh -> prepare($sql2);
$query2->execute();
$results2=$query2->fetchAll(PDO::FETCH_OBJ);
$cnt2=$query2->rowCount();
                                        ?>
```

```
                                        <h4><?java echo htmlentities($cnt2);?></h4>
                                    </div>
                                </div>
                            </div>
                            <div class="col-md-3 four-grid">
                                <div class="four-wthree">
                                    <div class="icon">
                                        <i class="glyphicon glyphicon-briefcase" aria-
hidden="true"></i>
                                    </div>
                                    <div class="four-text">
                                        <h3>Toatal packages</h3>

                        <?java $sql3 = "SELECT PackageId from tbltourpackages";
$query3= $dbh -> prepare($sql3);
$query3->execute();
$results3=$query3->fetchAll(PDO::FETCH_OBJ);
$cnt3=$query3->rowCount();
                            ?>
                                        <h4><?java echo htmlentities($cnt3);?></h4>
                                    </div>
                                </div>
                            </div>
                                <div class="clearfix"></div>
                        </div>
                <div class="four-grids">
                            <div class="col-md-3 four-grid">
                                <div class="four-w3ls">
                                    <div class="icon">
                                        <i class="glyphicon glyphicon-folder-open" aria-
hidden="true"></i>
                                    </div>
                                    <div class="four-text">
                                        <h3>Issues Riaised</h3>
```

```
                                                                    <?java $sql5 =
"SELECT id from tblissues";
$query5= $dbh -> prepare($sql5);
$query5->execute();
$results5=$query5->fetchAll(PDO::FETCH_OBJ);
$cnt5=$query5->rowCount();
                                        ?>
                                                <h4><?java echo htmlentities($cnt5);?></h4>
                                            </div>
                                        </div>
                                    </div>
                                    <div class="clearfix"></div>
                                </div>
<!--//four-grids here-->
<div class="inner-block">
</div>
<!--inner block end here-->
<!--copy rights start here-->
<?java include('includes/footer.java');?>
</div>
</div>
                        <!--/sidebar-menu-->
                            <?java include('includes/sidebarmenu.java');?>
                                                  <div class="clearfix"></div>
                                                </div>
                                                <script>
                                                var toggle = true;
                                                $(".sidebar-icon").click(function() {
                                                  if (toggle)
                                                  {
                                                        $(".page-container").addClass("sidebar-
collapsed").removeClass("sidebar-collapsed-back");
                                                        $("#menu span").css({"position":"absolute"});
                                                  }
```

```
                                                    else
                                                    {
                                                            $(".page-container").removeClass("sidebar-
collapsed").addClass("sidebar-collapsed-back");
                                                            setTimeout(function() {
                                                              $("#menu span").css({"position":"relative"});
                                                            }, 400);
                                                    }
                                                                                    toggle = !toggle;
                                                                            });
                                                </script>
<!--js -->
<script src="js/jquery.nicescroll.js"></script>
<script src="js/scripts.js"></script>
<!-- Bootstrap Core JavaScript -->
  <script src="js/bootstrap.min.js"></script>
  <!-- /Bootstrap Core JavaScript -->
<!-- morris JavaScript -->
<script src="js/raphael-min.js"></script>
<script src="js/morris.js"></script>
<script>
        $(document).ready(function() {
                //BOX BUTTON SHOW AND CLOSE
          jQuery('.small-graph-box').hover(function() {
                jQuery(this).find('.box-button').fadeIn('fast');
          }, function() {
                jQuery(this).find('.box-button').fadeOut('fast');
          });
          jQuery('.small-graph-box .box-close').click(function() {
                jQuery(this).closest('.small-graph-box').fadeOut(200);
                return false;
          });
           //CHARTS
           function gd(year, day, month) {
```

```
                return new Date(year, month - 1, day).getTime();
            }
            graphArea2 = Morris.Area({
                element: 'hero-area',
                padding: 10,
behaveLikeLine: true,
gridEnabled: false,
gridLineColor: '#dddddd',
axes: true,
resize: true,
smooth:true,
pointSize: 0,
lineWidth: 0,
fillOpacity:0.85,
                data: [
                    {period: '2014 Q1', iphone: 2668, ipad: null, itouch: 2649},
                    {period: '2014 Q2', iphone: 15780, ipad: 13799, itouch: 12051},
                    {period: '2014 Q3', iphone: 12920, ipad: 10975, itouch: 9910},
                    {period: '2014 Q4', iphone: 8770, ipad: 6600, itouch: 6695},
                    {period: '2015 Q1', iphone: 10820, ipad: 10924, itouch: 12300},
                    {period: '2015 Q2', iphone: 9680, ipad: 9010, itouch: 7891},
                    {period: '2015 Q3', iphone: 4830, ipad: 3805, itouch: 1598},
                    {period: '2015 Q4', iphone: 15083, ipad: 8977, itouch: 5185},
                    {period: '2016 Q1', iphone: 10697, ipad: 4470, itouch: 2038},
                    {period: '2016 Q2', iphone: 8442, ipad: 5723, itouch: 1801}
                ],
                lineColors:['#ff4a43','#a2d200','#22beef'],
                xkey: 'period',
  redraw: true,
  ykeys: ['iphone', 'ipad', 'itouch'],
  labels: ['All Visitors', 'Returning Visitors', 'Unique Visitors'],
                pointSize: 2,
                hideHover: 'auto',
                resize: true
```

```
                    });
            });
            </script>
</body>
</html>
<?java } ?>
<?java
session_start();
include('includes/config.java');
if(isset($_POST['login']))
{
$uname=$_POST['username'];
$password=md5($_POST['password']);
$sql ="SELECT UserName,Password FROM admin WHERE UserName=:uname and Password=:password";
$query= $dbh -> prepare($sql);
$query-> bindParam(':uname', $uname, PDO::PARAM_STR);
$query-> bindParam(':password', $password, PDO::PARAM_STR);
$query-> execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
$_SESSION['alogin']=$_POST['username'];
echo "<script type='text/javascript'> document.location = 'dashboard.java'; </script>";
} else{
            echo "<script>alert('Invalid Details');</script>";
}
}
?>
<!DOCTYPE HTML>
<html>
<head>
<title>TMS | Admin Sign in</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- Bootstrap Core CSS -->
<link href="css/bootstrap.min.css" rel='stylesheet' type='text/css' />
<!-- Custom CSS -->
<link href="css/style.css" rel='stylesheet' type='text/css' />
<link rel="stylesheet" href="css/morris.css" type="text/css"/>
<!-- Graph CSS -->
<link href="css/font-awesome.css" rel="stylesheet">
<link rel="stylesheet" href="css/jquery-ui.css">
<!-- jQuery -->
<script src="js/jquery-2.1.4.min.js"></script>
<!-- //jQuery -->
<link href='//fonts.googleapis.com/css?family=Roboto:700,500,300,100italic,100,400' rel='stylesheet' type='text/css'/>
<link href='//fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>
<!-- lined-icons -->
<link rel="stylesheet" href="css/icon-font.min.css" type='text/css' />
<!-- //lined-icons -->
</head>
<body>
	<div class="main-wthree">
	<div class="container">
	<div class="sin-w3-agile">
		<h2>Sign In</h2>
		<form  method="post">
			<div class="username">
				<span class="username">Username:</span>
				<input type="text" name="username" class="name" placeholder="" required="">
				<div class="clearfix"></div>
			</div>
			<div class="password-agileits">
				<span class="username">Password:</span>
				<input type="password" name="password" class="password" placeholder=""
```

```
required="">
                                        <div class="clearfix"></div>
                                </div>
                                <div class="login-w3">
                                        <input type="submit" class="login" name="login" value="Sign In">
                                </div>
                                <div class="clearfix"></div>
                        </form>
                                        <div class="back">
                                                <a href="../index.java">Back to home</a>
                                        </div>
                </div>
                </div>
                </div>
</body>
</html><?java
session_start();
error_reporting(0);
include('includes/config.java');
if(strlen($_SESSION['alogin'])==0)
        {
header('location:index.java');
}
else{
        // code for cancel
if(isset($_REQUEST['bkid']))
        {
$bid=intval($_GET['bkid']);
$status=2;
$cancelby='a';
$sql = "UPDATE tblbooking SET status=:status,CancelledBy=:cancelby WHERE  BookingId=:bid";
$query = $dbh->prepare($sql);
$query -> bindParam(':status',$status, PDO::PARAM_STR);
$query -> bindParam(':cancelby',$cancelby , PDO::PARAM_STR);
```

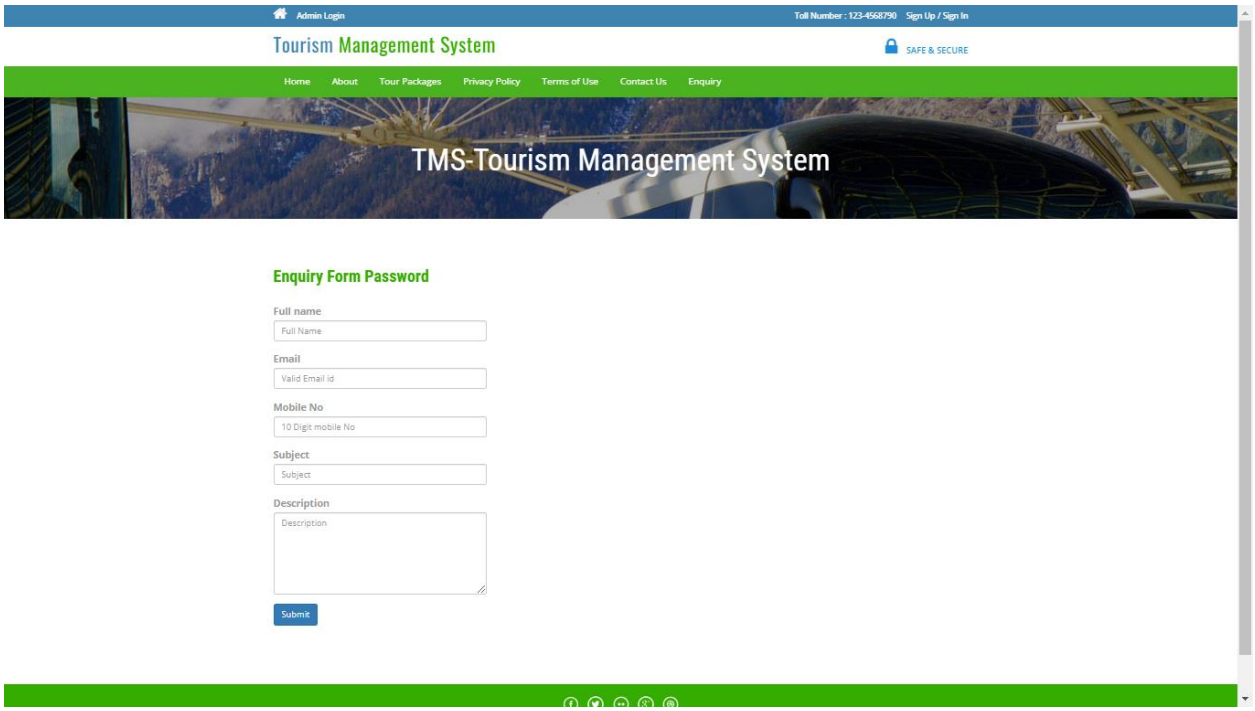
```
$query-> bindParam(':bid',$bid, PDO::PARAM_STR);
$query -> execute();
$msg="Booking Cancelled successfully";
}
if(isset($_REQUEST['bckid']))
        {
$bcid=intval($_GET['bckid']);
$status=1;
$cancelby='a';
$sql = "UPDATE tblbooking SET status=:status WHERE BookingId=:bcid";
$query = $dbh->prepare($sql);
$query -> bindParam(':status',$status, PDO::PARAM_STR);
$query-> bindParam(':bcid',$bcid, PDO::PARAM_STR);
$query -> execute();
$msg="Booking Confirm successfully";
}
        ?>
<!DOCTYPE HTML>
<html>
<head>
<title>TMS | Admin manage Bookings</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); },
false); function hideURLbar(){ window.scrollTo(0,1); } </script>
<link href="css/bootstrap.min.css" rel='stylesheet' type='text/css' />
<link href="css/style.css" rel='stylesheet' type='text/css' />
<link rel="stylesheet" href="css/morris.css" type="text/css"/>
<link href="css/font-awesome.css" rel="stylesheet">
<script src="js/jquery-2.1.4.min.js"></script>
<link rel="stylesheet" type="text/css" href="css/table-style.css" />
<link rel="stylesheet" type="text/css" href="css/basictable.css" />
<script type="text/javascript" src="js/jquery.basictable.min.js"></script>
<script type="text/javascript">
```

```
    $(document).ready(function() {
      $('#table').basictable();
      $('#table-breakpoint').basictable({
        breakpoint: 768
      });
      $('#table-swap-axis').basictable({
        swapAxis: true
      });
      $('#table-force-off').basictable({
        forceResponsive: false
      });
      $('#table-no-resize').basictable({
        noResize: true
      });
      $('#table-two-axis').basictable();
      $('#table-max-height').basictable({
        tableWrapper: true
      });
    });
</script>
<link href='//fonts.googleapis.com/css?family=Roboto:700,500,300,100italic,100,400' rel='stylesheet'
type='text/css'/>
<link href='//fonts.googleapis.com/css?family=Montserrat:400,700' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="css/icon-font.min.css" type='text/css' />
  <style>
                        .errorWrap {
    padding: 10px;
    margin: 0 0 20px 0;
    background: #fff;
    border-left: 4px solid #dd3d36;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
.succWrap{
```

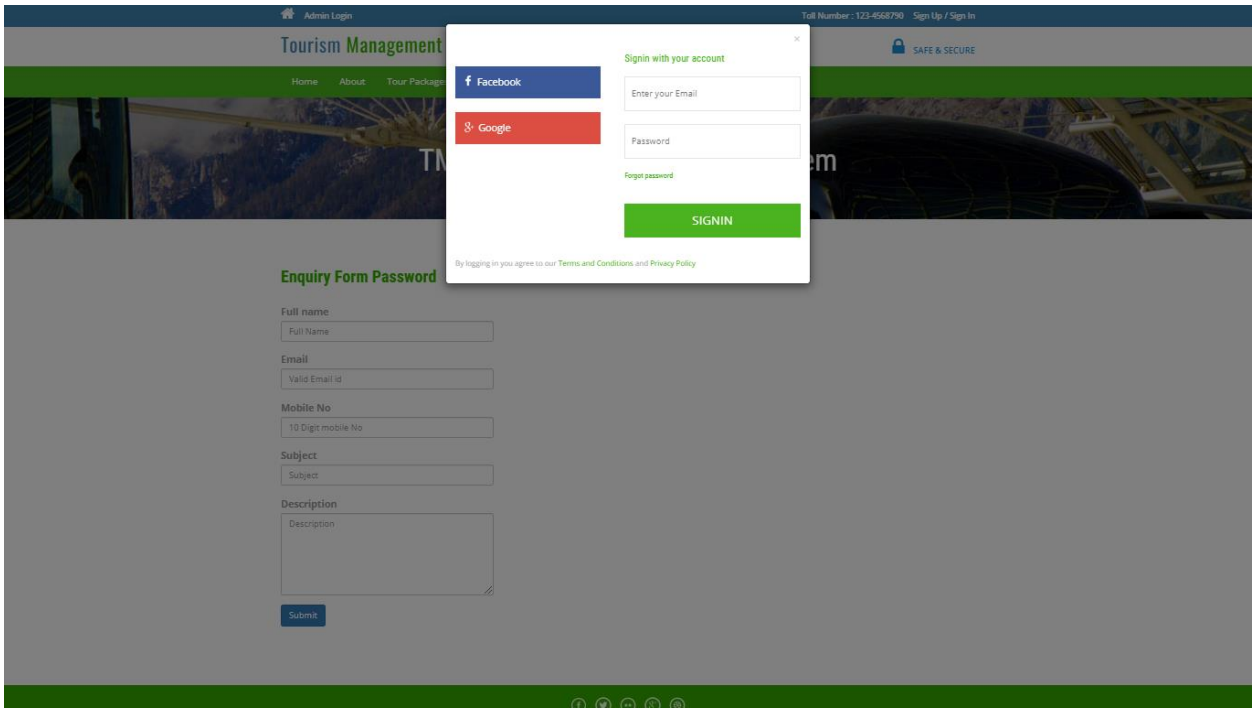
```
    padding: 10px;
    margin: 0 0 20px 0;
    background: #fff;
    border-left: 4px solid #5cb85c;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
                    </style>
</head>
<body>
  <div class="page-container">
  <!--/content-inner-->
<div class="left-content">
            <div class="mother-grid-inner">
        <!--header start here-->
                                    <?java include('includes/header.java');?>
                                        <div class="clearfix"> </div>
                                    </div>
<!--heder end here-->
<ol class="breadcrumb">
            <li class="breadcrumb-item"><a href="index.html">Home</a><i class="fa fa-angle-right"></i>Manage
Bookings</li>
        </ol>
<div class="agile-grids">
                                    <!-- tables -->
                                    <?java if($error){?><div class="errorWrap"><strong>ERROR</strong>:<?java
echo htmlentities($error); ?> </div><?java }
                                    else if($msg){?><div class="succWrap"><strong>SUCCESS</strong>:<?java
echo htmlentities($msg); ?> </div><?java }?>
                                    <div class="agile-tables">
                                            <div class="w3l-table-info">
                                              <h2>Manage Bookings</h2>
                                                <table id="table">
                                                        <thead>
```

D. Sample Input

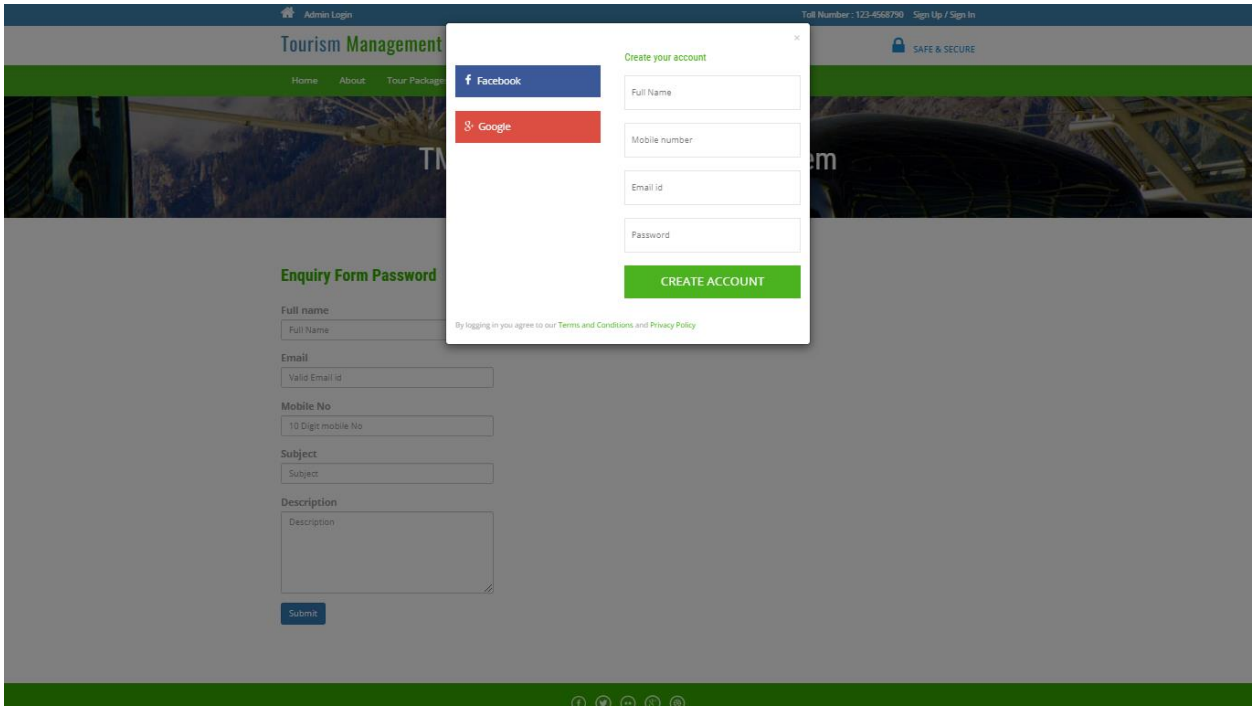
Input of enquiry registration



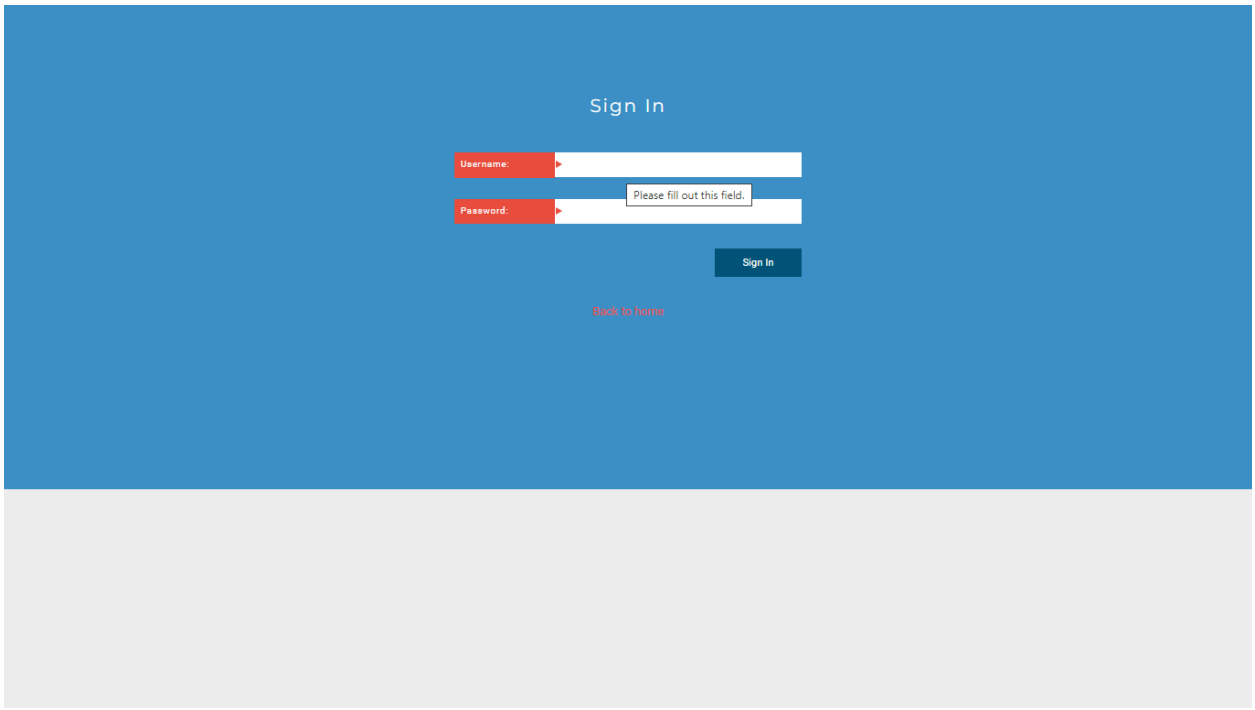
User login page with user input



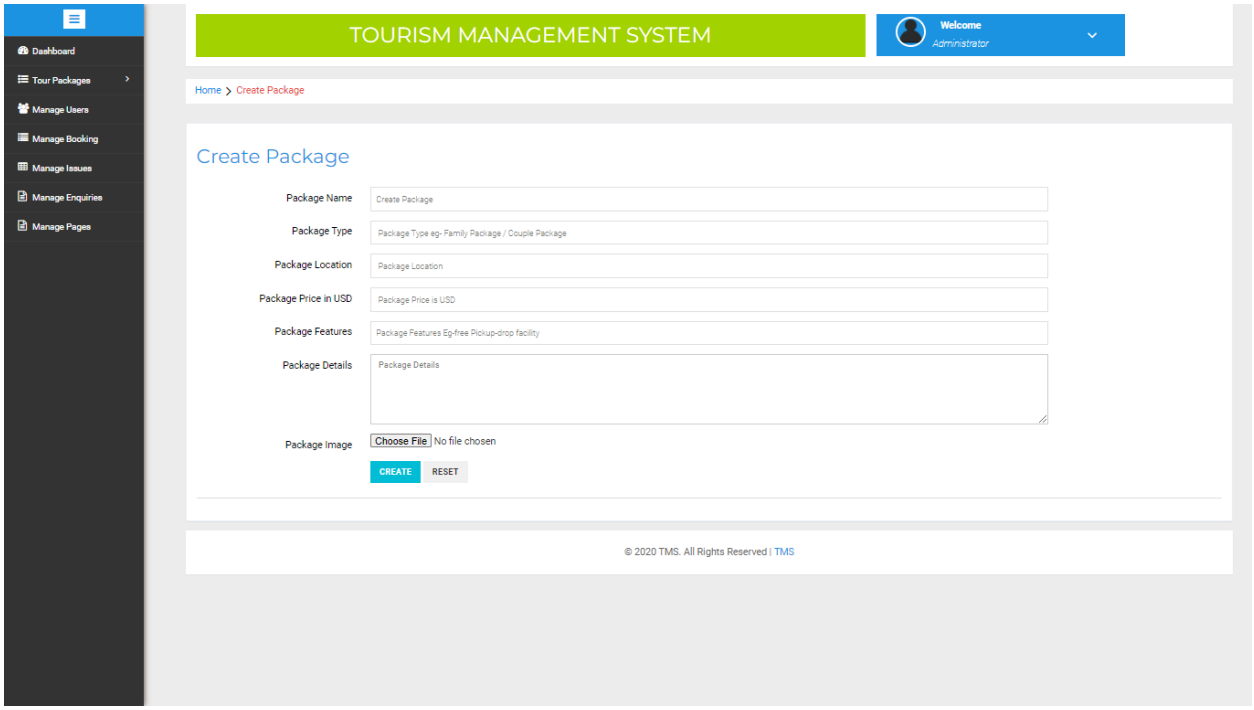
Input of create user account



Admin login page

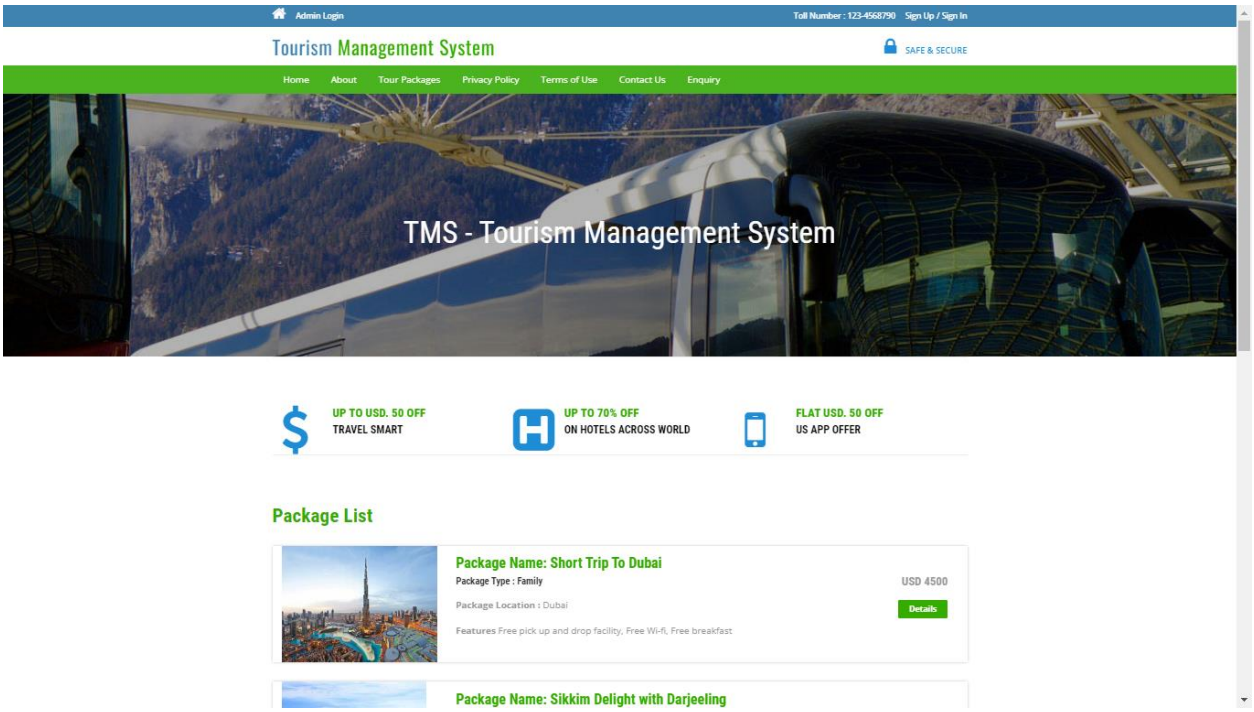


Input of create package details

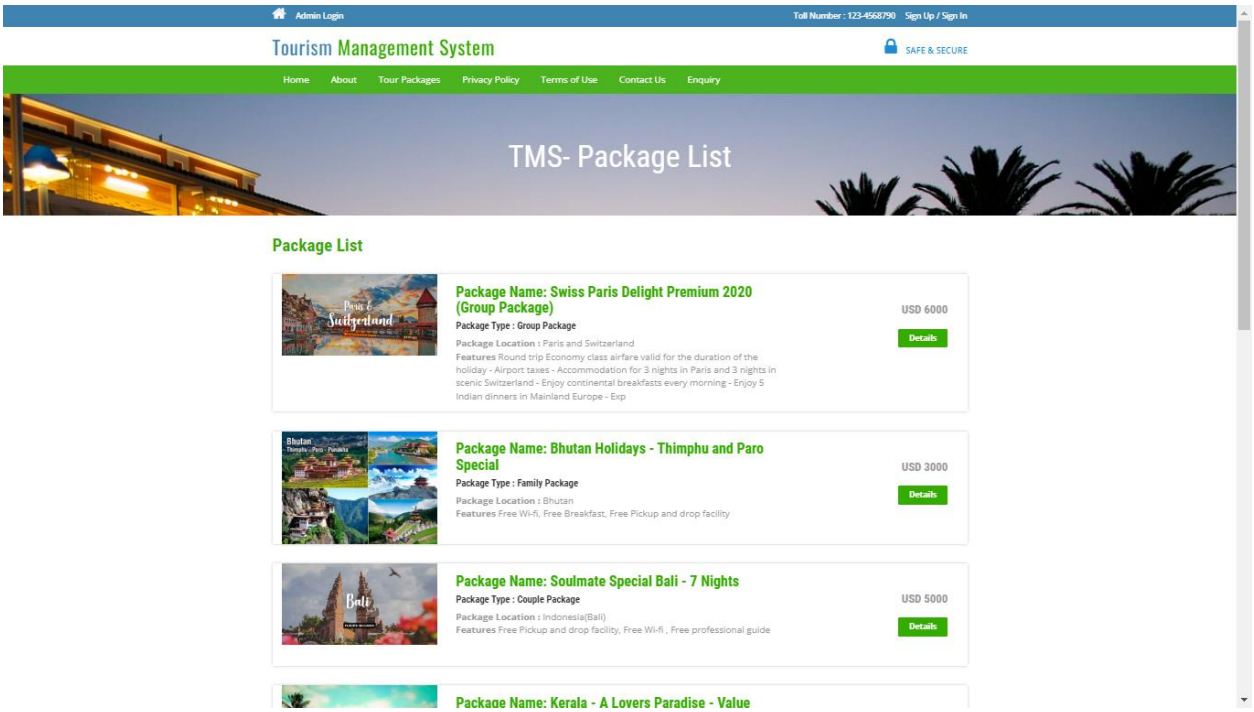


E. Sample Output

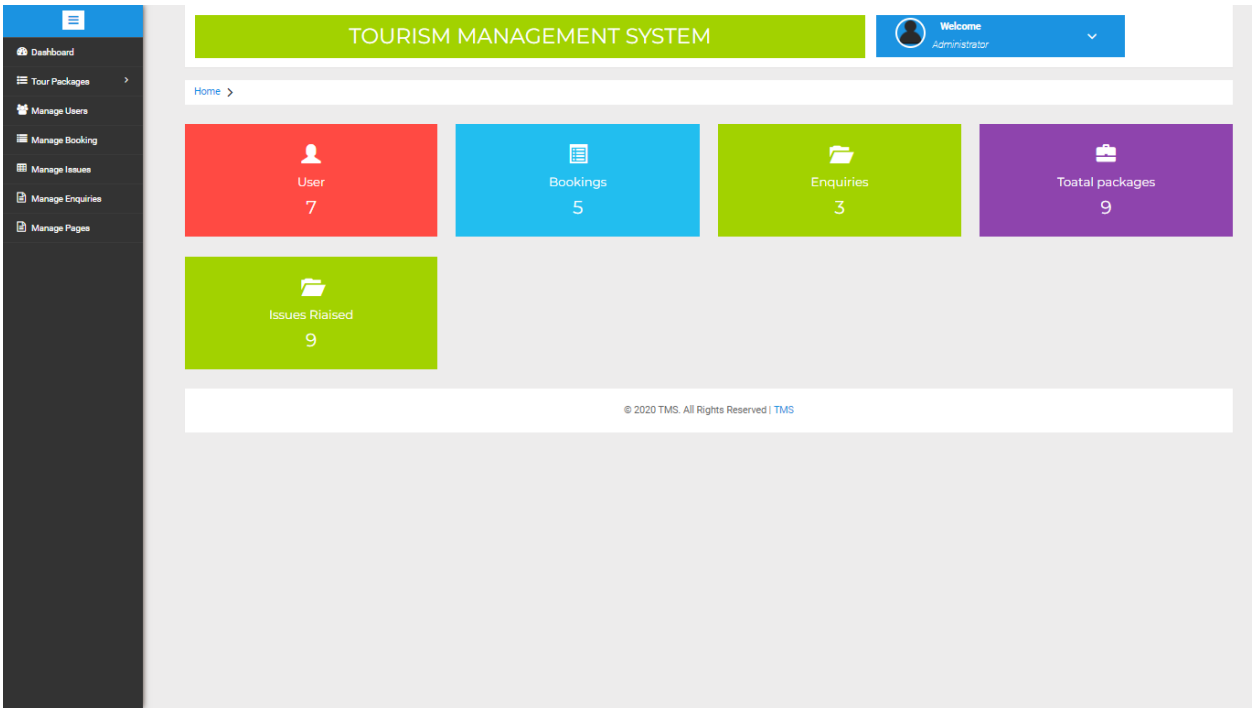
Home page



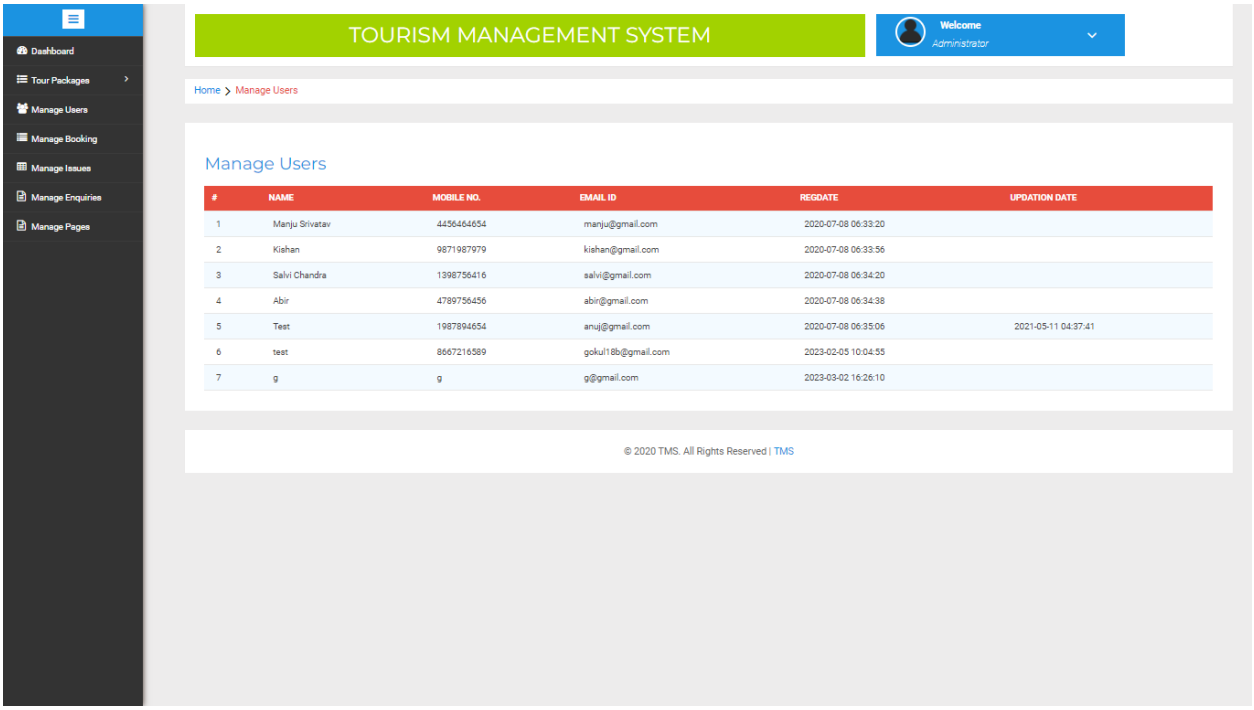
Output of Package list details



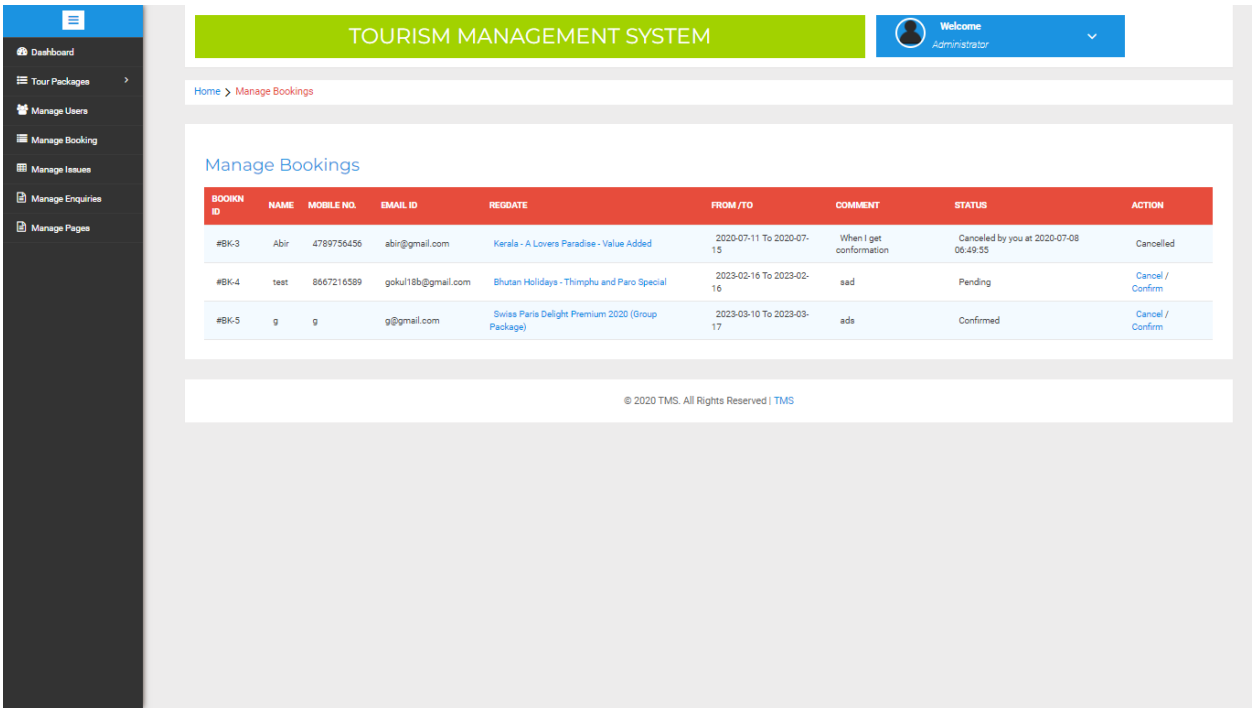
Output of dashboard



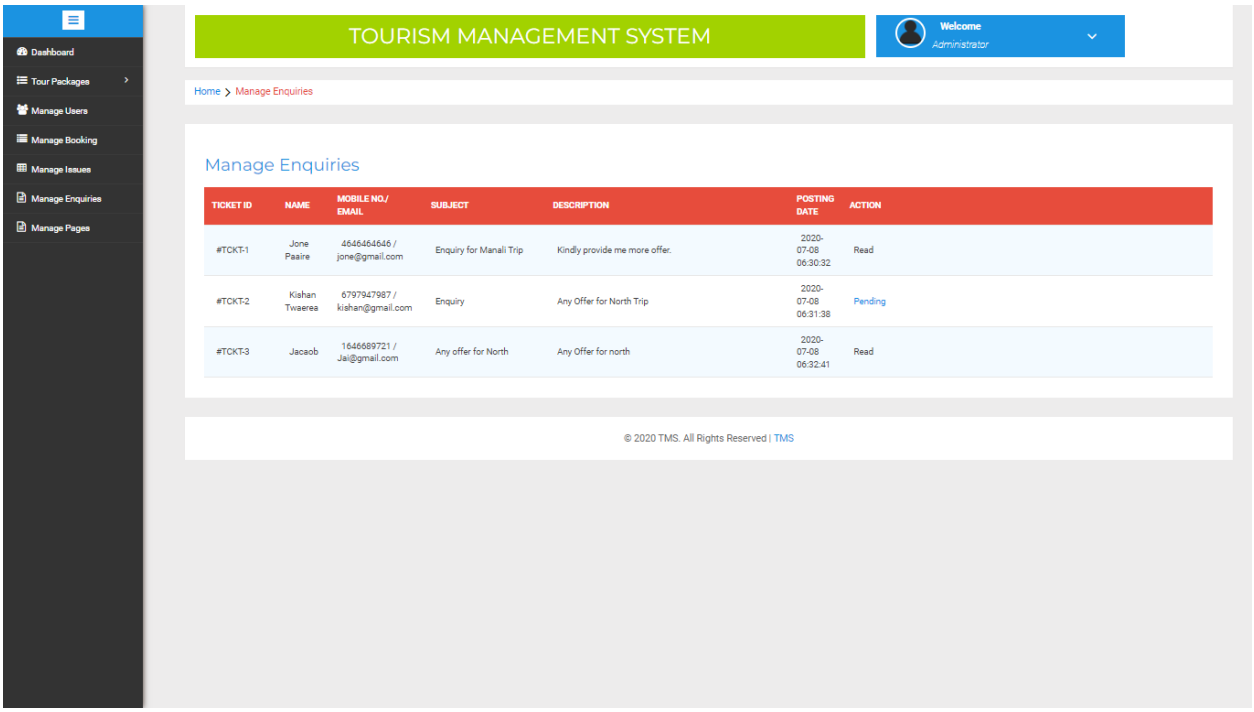
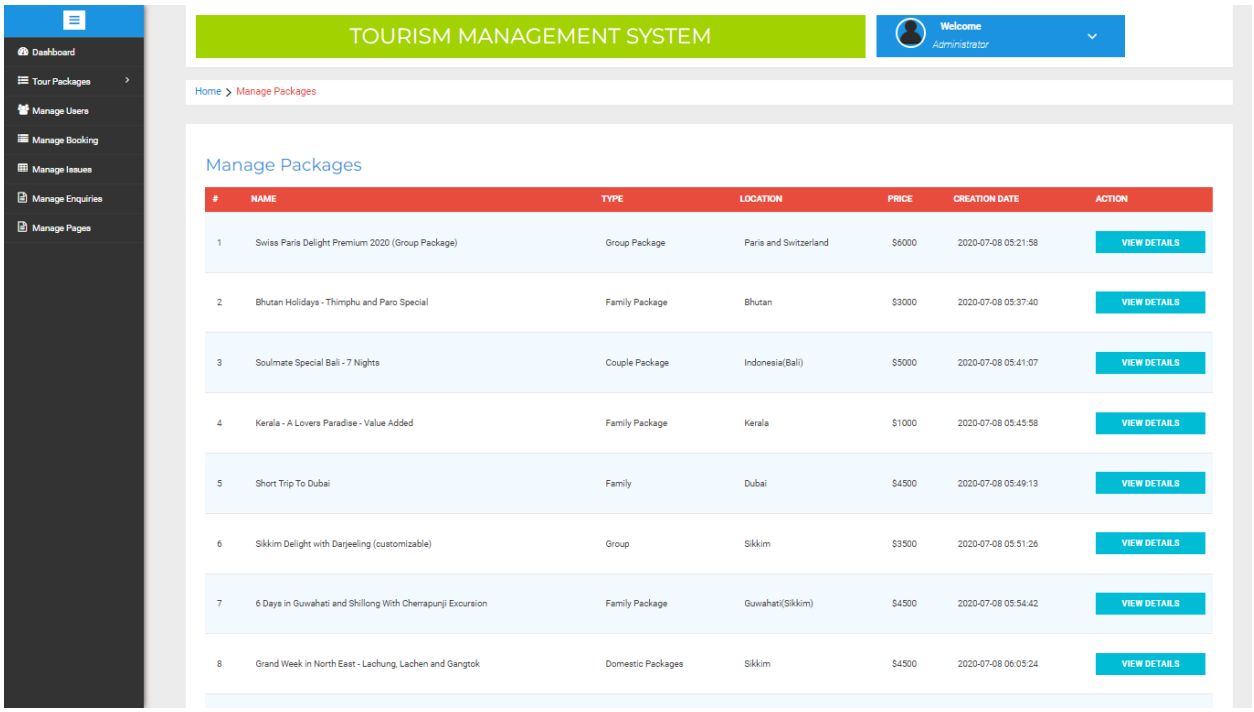
Output of user details



Output of booking details



Output of Enquiery



Output of package details