

CHAPTER NO: 1

PROJECT SYNOPSIS

1.1 PROJECT ABSTRACT:

This project mainly concentrates on tours and travels the maintenance of this software module such as Booking, Facilities and Services, Tour Packages, Tour Booking Details and so on. This project is completely well designed and managing to store the data base records in the SQL server and also developed for the maintaining day-to-day transactions detailed of Tour booking information for each Vehicle till the next day transaction starts. This software is developed for current booking only and necessary for holding the data or information of Vehicle with detailed information of each Customer.

1.2 PROJECT INTRODUCTION

The developments of this project are managing the details of tours with travelling of the different types of places and to take the details of the Passenger, Package reservation, Tour details, Car Rentals etc. All these above modules are here the information is to be stored in a SQL Server as back-end tool of database and modules are well graphical formations of forms design in visual basic .net software as a front-end tool. But all these modules should be stored, updated and deleting the records can be easily done with the permission of the project administrator. The main aim of this project is to manage the records such as Tour Details, Customer Details, Vehicle Details, Facilities and Services, Tour Packages, Tour Booking, Car Booking and so on. Also maintaining day-to-day transactions detailed of Tour booking information for each Vehicle till the next day transaction starts and the data stored in a data base can be manipulated to suit our needs. We can get all the details easily. Every module takes the data accurately and automatic generation of unique code should be done in program code so there is no duplications of data entry. Every user can operate the project modules easily and friendly to use all the data entry with validation of each attribute values of data.

1.23 SCOPE OF THE PROJECT:

This project is mainly developed on the package tour details booking in this automation software tool such as Customer Details, Vehicle Details, Facilities and Services, Tour

Packages, Tour Booking Details, etc. The scope of these projects is well designed and sample coding with validations and also user-friendly project.

1.4 LIST OF THE PROJECT MODULES:

1. Package Details
2. Package Reservation
3. Customer Details
4. Vehicle Details
5. Facilities and services
6. Tour Package
7. Tourism Places
8. Tour Booking Details
9. View Details
10. Ticket Cancellation
11. Final Report Generation

1.5 HARDWARE REQUIRMENTS:

Processor	Intel I3 2.30 GHz
RAM	4GB DDR RAM
Hard Disk	160GB Seagate Hard Disk
Input Devices	Standard Keyboard and Mouse
Output Device	15' LED Monitor

1.6 SOFTWARE REQUIRMENTS:

Operating System	Windows 11 professional
Front end tool	VB.NET 2010
Back-end tool	SQL SERVER 2010

CHAPTER NO: 2

PROJECT INTRODUCTION

We developed the software project called Tours & Travels and it is used to maintain the records such as Booking details, Package details, Cancellation details, Records of all package reservation, Car rentals details are managed in this module. This project is developed using VB.Net 2010 as front-end tool and MS SQL Server 2010 as back-end tool. The developments of this project are managing the details of tours with travelling of the different types of places and to take the details of the Customer, vehicle booking, selection of tour package. The main aim of this project is to manage the records such as Tour Details, Customer Details, Vehicle Details, Facilities and Services, Tour Packages, Tour Booking, Car Booking and so on. Also maintaining day-to-day transactions detailed of Tour booking information for each Vehicle till the next day transaction starts and the data stored in a data base can be manipulated to suit our needs. Every user can operate the project modules easily and friendly to use all the data entry with validation of each attribute values of data.

This project is mainly developed on the package tour details booking in this automation software tool such as Customer Details, Vehicle Details, Facilities and Services, Tour Packages, Tour Booking Details, etc. The scope of these projects is well designed and sample coding with validations and also user-friendly project.

2.1 PROJECT OBJECTIVE

The main objective of the project is to design innovative software, which deals with the “***Tours and Travels***”. The motto of the project is to simplify the job of the administrative people and to render a user-friendly software package tool. To Develop the records in this project like Tour Details, Customer Details, Vehicle Details, Facilities and Services, Tour Packages, Package Reservation, Car Booking and all the records and transactions. Also maintaining day-to-day transactions detailed of Tour booking information for each Vehicle till the next day transaction starts and the data stored in a data base can be manipulated to suit our needs.

2.2 Details of Project Modules

2.2.1 Tour Details:

This Module contains the details of all the packages such as North India Package, Goa Package, Devotional Package, South India Package regarding the amount, places, days & nights, accommodation, food charge, travelling expenses, miscellaneous expenses and so on. The above Details are stored in this module.

2.2.2 Bookings:

This Module consist the details to be entered to make reservations for all the packages such as North India Package, Goa Package, Devotional Package, South India Package & Car Rentals.

2.2.3 Search Details:

This Module contains all the details of all Passengers and their packages which can be retrieved by entering PassengerID in this module.

2.2.4 Modify Details:

This Module consist the details of all Passengers and their packages and Car rentals which can be modified and saved in the database.

2.2.5 Show Records:

This Module shows the data of reservation of packages. The data is saved and stored in particular package records of which the reservation is done. Each package stores its separate records and displayed in this module.

2.2.6 Cancellation Details:

The reservation of packages can be cancelled using this module and those cancelled records are stored in this module which can be retrieved by using this module.

2.2.7 Print Ticket:

This Module is used to print the reserved ticket of passenger by selecting the PassengerID after booking or modifying the record. So, this module is used to store all the details such as his name, gender, address, contact number, email id, date of booking, date of departure, package type and its details and amount.

2.3 Need for computerization

The increasing complexity of services and hence the large quantities of information necessary to be maintained by modern technology has made the computerization of the front office system a necessity, and this system is specifically designed for the ownerin mind as the end user.

- Easy to use.
- Provides more functionality.
- Provides more information about cases.
- Quick access to information.
- More versatile.

The advantages of the proposed system as follows:

- To reduce the workload.
- To reduce the processing time.
- To view the seat arrangement to maintain reliable and update information.
- Easy accessibility to
- Computerized quick report for every database.

CHAPTER NO: 3

HARDWARE & SOFTWARE CONFIGURATIONS

This software is a modern and latest technology system specifications used for development and inserted well graphics images, navigations, tools, object components in visual studio 2010 application software as a frontend in this project work and also current market hardware configurations used in this version of the software. The minimum hardware configured using 2GB RAM is required. But all these records are stored in a database package is SQL server 2010 as a backend toolin this software. Here there is no problem for software and hardware configurations another 10 years definitely, so that we are preferable this software and hardware configurations.

3.1 HARDWARE REQUIREMENT

Processor	:	Intel Core Duo 2.0 GHz or more
RAM	:	1 GB or More
Hard disk	:	80GB or more
Monitor	:	15" CRT or LCD monitor
Keyboard	:	Normal or Multimedia
Mouse	:	Compatible mouse

3.2 SOFTWARE REQUIREMENT

Operating System	:	Microsoft Windows 11
Front End Tool	:	VB .NET 2010
Back End Tool	:	Microsoft SQL Server 2010

CHAPTER: 4

SOFTWARE REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) is a document that clearly and precisely specifies each and every essential requirement for the software product, as well as the external interfaces to hardware & firmware. Each requirement should be defined so that it can be verified by a method such as inspection, demonstration, analysis and testing. There are number of desirable properties that a SRS should possess. In particular, the requirements documents should be

- Correct
- Complete
- Consistent
- Unambiguous
- Functional
- Verifiable
- Traceable
- Easily changed

An incorrect or incomplete set of requirements can result in a software product that satisfies but does not satisfy customer needs. SRS should be functional in nature; i.e., they should describe what is required without implying how the system will need its requirements.

The purpose of this document is to prescribe the software requirements for “Tours & Travels” to be developed. In this software we analyze the essence of maintaining, modifying, and removing the account and its data. The software requirement specification is completely done by the following project modules. Each and every module designed in VB.Net Front end tool and Data base records stored in SQL server.

4.1 Tour Details:

This Module contains the details of all the packages such as North India Package, Goa Package, Devotional Package, South India Package regarding the amount, places, days & nights, accommodation, food charge, travelling expenses, miscellaneous expenses and so on.

The above Details are stored in this module.

4.2 Bookings:

This Module consist the details to be entered to make reservations for all the packages such as North India Package, Goa Package, Devotional Package, South India Package & Car Rentals.

4.3 Search Details:

This Module contains all the details of all Passengers and their packages which can be retrieved by entering PassengerID in this module.

4.4 Modify Details:

This Module consist the details of all Passengers and their packages and Car rentals which can be modified and saved in the database.

4.5 Show Records:

This Module shows the data of reservation of packages. The data is saved and stored in particular package records of which the reservation is done. Each package stores its separate records and displayed in this module.

4.6 Cancellation Details:

The reservation of packages can be cancelled using this module and those cancelled records are stored in this module which can be retrieved by using this module.

4.7 Final Report & Print Ticket:

This Module is used to print the reserved ticket of passenger by selecting the PassengerID after booking or modifying the record. So, this module is used to store all the details such as his name, gender, address, contact number, email id, date of booking, date of departure, package type and its details and amount.

CHAPTER: 5

INTRODUCTION TO SQL SERVER AND BACK END TOOL

A new generation of applications is on the horizon for businesses and consumers. This is driven by market changes such as a growing mobile workforce and its need for productivity even when not “connected” and an explosive growth of new devices such as personal digital assistants (PDAs), portable music players, and gaming consoles, creating exciting possibilities for new applications. These new applications place new demands on the data platform on which they are developed.

To meet the needs of this changing application landscape, Microsoft introduces its newest edition of the SQL Server database product line, SQL Server 2010 Compact Edition — a maintenance-free, compact embedded database for single-user client applications for all Windows Platforms, including Tablet PCs, Pocket PCs, Smart Phones and Desktops. Compact Edition is available to deploy and redistribute free of charge.

This white paper outlines the current business and consumer environment and the key factors that influence the need for this new generation of applications. It describes how SQL Server Compact Edition can be used to satisfy the data platform demands of these applications.

Business Environment and Customer Challenges:

Evolving business and consumer needs are placing new demands on the next generation of applications. In the business sphere, more and more employees or information workers are mobile and are seeing the need to run their traditional desktop applications on mobile devices like PDAs and Tablet PCs. For example, sales personnel require Customer Relationship Management (CRM) applications on their mobile devices; field service employees need to check product specifications and perform online ordering from mobile devices, and so on. In addition, these mobile information workers cannot rely on being connected to the corporate network at all times, but need to have the same rich and complete application experience in the “disconnected” or “occasionally connected” state, as they would have in the “connected” state. A classic example of this is the offline e-mail experience presented by Microsoft Office Outlook, where users can work on their locally stored e-mails in a disconnected state and synchronize with the server when connectivity is available. Information workers also expect a seamless integrated experience between multiple devices. For example, one may start entering

data into an expense reporting application on a PDA and continue working on the same application from a desktop at a later point, starting from where he or she left off on the PDA.

Another growing trend in businesses today is to gain competitive advantage by gaining insights from a plethora of data that is collected every day. This data is being gathered at the “edges” of the enterprise – for example: inventory information gathered through scanning radio frequency identification (RFID) tags at a warehouse – and needs to be aggregated with data from other edge locations at a central site for analysis. Similarly, the mobile workforce that works at the edges of the enterprise needs relevant and current business information from the central data repository.

Meanwhile, information technology (IT) departments that develop and deploy these line-of-business (LOB) applications in their businesses continue to grapple with the challenge of providing robust IT services that are secure and reliable in spite of constantly falling budgets. IT departments continue to look for ways to develop new applications rapidly by leveraging existing skills and tools and to find ways to easily deploy and manage end-user applications, which helps to keep costs down.

For consumers, rich personal applications continue to proliferate and evolve on desktops and home devices, such as gaming consoles and digital video recorders (DVRs). These applications are becoming more and more connected as vendors sell additional personalized services through the Internet. For example, a vendor that sells a tax application can offer services to consumers to securely store their year-end data online for any future need. Similarly, vendors can offer enhanced end-user experiences with their client applications using this service model. For example, an online streaming audio/video service that automatically downloads images of the album or video cover can enhance the end-user experience. Staying in touch with the customer through an Internet service can also help vendors track user preferences and better target their software offerings.

Rich consumer applications also continue to grow rapidly in mobile phones and in the convergent devices arena. Applications such as calendaring that were in the business realm are now also increasingly being required in personal devices.

As application paradigms change, application developers have new requirements for the database platform on which these applications are built. In particular, there is a growing need for local-store databases that are:

- **Embeddable** to provide better application security, performance, and ease of distribution and deployment.
- **Lightweight and compact** to preserve system resources, especially in devices where processor and memory is premium.
- **Componentizable** to allow application vendors to choose essential functionality only, this keeps application size to a minimum.
- **Supported on multiple devices** so the same application code can be used to build applications on multiple devices and platforms.
- **Capable of providing automatic synchronization** to allow end-users to work with most recent data in an offline application experience and to push updates to a central location.
- **Built with advanced security** to protect against threats such as stolen mobile devices.

In light of the changing needs of the new class of applications, Microsoft introduces its newest edition of the SQL Server database product line, SQL Server 2010 Compact Edition. It is an evolution of the SQL Server Mobile Edition technology. While the latter was used exclusively for mobile devices, its capabilities are now being leveraged to create maintenance-free, compact (less than 2 MB disk footprint) embedded databases for single-user client applications for all Microsoft Windows platforms, including Tablet PCs, Pocket PCs, smart phones, and desktops, to enable the scenarios discussed earlier in the paper.

SQL Server Compact Edition also enables an application to scale up, either via a robust synchronization with other SQL Server Editions, or by moving to a higher edition of SQL Server.

How SQL Server Compact Edition Can Help

SQL Server Compact Edition addresses the new application development challenges by providing for faster results and increased productivity, better decision-making support and a trusted platform.

FASTER RESULTS AND INCREASED PRODUCTIVITY

Developers and administrators can achieve faster results with SQL Server Compact Edition.

DEVELOPERS

SQL Server Compact Edition is tightly integrated with Microsoft Visual Studio 2010 as well as the .NET Framework. Developers can readily use their existing knowledge of developing database applications with Visual Studio 2010 to build SQL Server Compact Edition based applications, which enables them to be instantly productive. SQL Server Compact Edition supports a familiar structured query language (SQL) syntax and common programming model, such as ADO.NET. A significant productivity advantage for developers is that they can use the same application code to build the application for multiple devices and platforms. SQL Server Compact Edition applications can also be seamlessly migrated from mobile and desktop scenarios to work with other editions of SQL Server 2010 in multi-user or server scenarios.

ADMINISTRATORS

As the database is embedded in the application, little or no management is required. SQL Server Compact Edition can be included in the application package and deployed using the Click Once deployment feature of Visual Studio 2010. Alternatively, this lightweight data store can be downloaded quickly. SQL Server Compact Edition can be configured to keep up-to-date with the Microsoft Update services, so administrators will not need to bother with complicated maintenance plans. However, the user needs to have administrative privileges to install SQL Server Compact Edition on a tablet PC or desktop. An administrator can use SQL Server Management Studio (including Express Edition) to administer the database.

BETTER DECISIONS

With SQL Server Compact Edition, together with SQL Server 2010 business intelligence capabilities, you can extend your business insights further across the organization to data sources in distributed and mobile applications. By enabling synchronization to a central database, data in such applications can be captured, analyzed by SQL Server Analysis Services, and reported by SQL Server Reporting Services. The Report Viewer Controls available in Visual Studio 2010 can be used in applications to provide rich reports based on data from SQL Server Compact Edition. With SQL Server Compact Edition, you can deliver the right data, at the right place and time, in the right form factor.

TRUSTED PLATFORM

As a leading database platform, SQL Server 2010 provides an unsurpassed level of trust. The SQL Server product line has been in existence for more than 10 years and has millions of customers.

RELIABILITY

Being a member of the SQL Server 2010 family, SQL Server Compact Edition takes advantage of proven technologies. Originally built on SQL Server Mobile technology, SQL Server Compact Edition has additional capabilities to ensure that applications perform well and are secure. SQL Server Compact Edition includes “self-healing” and management technologies that are designed to handle “instant off” and dead battery scenarios in mobile devices.

SQL Server Compact Edition supports two synchronization technologies to ensure that data is reliably delivered, supporting updates on both the server and client end. Remote Data Access (RDA) is a synchronization technology that can be used when a lightweight synchronization solution is adequate (e.g. no conflict resolution). SQL Server Compact Edition also supports merge replication, which is ideal when a more sophisticated solution is required.

Security was a critical part of the design of SQL Server Compact Edition. Because it was designed into the product, organizations can trust the high security level for data protection in these ways:

- Synchronization uses 128-bit SSL for secure and reliable functionality, even through firewalls.

- SQL Server Compact Edition uses 128-bit RSA file-based encryption on devices for database file security.
- SQL Server Compact Edition provides an encrypted data format with password protection.
- SQL Server Compact Edition uses a single file format, enabling document-safe format.

BETTER OVER TIME

SQL Server scales up from a device to the data center, so as business requirements grow, SQL Server Compact Edition can easily scale up to other editions of SQL Server. Applications of all sizes are supported on the trusted SQL Server platform, including:

- ISV applications that run on mobile devices or a local desktop.
- Occasionally connected applications that need automatic synchronization.
- Small company, departmental, and workgroup scenarios.
- Web-based applications.

CONCLUSION

SQL Server 2010 Compact Edition rises to the challenge of enabling application vendors and IT organizations to create the next wave of applications. Whether it is providing a rich and complete desktop-like application experience to the mobile information workers who are occasionally connected or consumers who are demanding more from their desktop and device applications, SQL Server Compact Edition has a host of features and capabilities that enable it to effectively serve as a local data store for these new applications. Such features include high security, high performance, simple development and deployment, and automatic synchronization.

DETAILED DESIGN

Detailed design covers the study of the major entities involved in the system, their attributes and relationships, how they can be transformed into normalized tables and what is the dependency among the tables. While system definition is design oriented, detailed design is implementation oriented. By defining logical and physical structure of the database, detailed design guides the implementation phase.

Database schema

LOGIN TABLE

Column Name	Data Type	Allow Nulls
Username	Varchar (50)	Not Null
Password	Varchar (50)	Not Null

BOOKING TABLE

Column Name	Datatype	Allow Null
PassengerID	Varchar(50)	Not Null
Name	Varchar(50)	Not Null
Gender	Varchar(50)	Not Null
Contact	Varchar(50)	Not Null
EmailID	Varchar(50)	Not Null
Add1	Varchar(50)	Not Null
Add2	Varchar(50)	Not Null
Add3	Varchar(50)	Not Null
Date	Varchar(50)	Not Null
Days	Varchar(50)	Not Null
Total	Varchar(50)	Not Null
Places	Varchar(500)	Not Null
Package	Varchar(50)	Not Null

CAR RENTALS TABLE

Column Name	Datatype	Allow Null
PassengerID	Varchar(50)	Not Null
Name	Varchar(50)	Not Null
Gender	Varchar(50)	Not Null
Contact	Varchar(50)	Not Null
EmailID	Varchar(50)	Not Null
Add1	Varchar(50)	Not Null
Add2	Varchar(50)	Not Null
Add3	Varchar(50)	Not Null
Date	Varchar(50)	Not Null
Vehicle	Varchar(50)	Not Null
TotalKM	Varchar(50)	Not Null
Driverfees	Varchar(50)	Not Null
Amount	Varchar(50)	Not Null

CANCELLATION TABLE

Column Name	Datatype	Allow Null
PassengerID	Varchar(50)	Not Null
Name	Varchar(50)	Not Null
Gender	Varchar(50)	Not Null
Contact	Varchar(50)	Not Null
EmailID	Varchar(50)	Not Null
Add1	Varchar(50)	Not Null
Add2	Varchar(50)	Not Null
Add3	Varchar(50)	Not Null
Date	Varchar(50)	Not Null
Days	Varchar(50)	Not Null
Total	Varchar(50)	Not Null
DOR	Varchar(50)	Not Null
Refund	Varchar(50)	Not Null

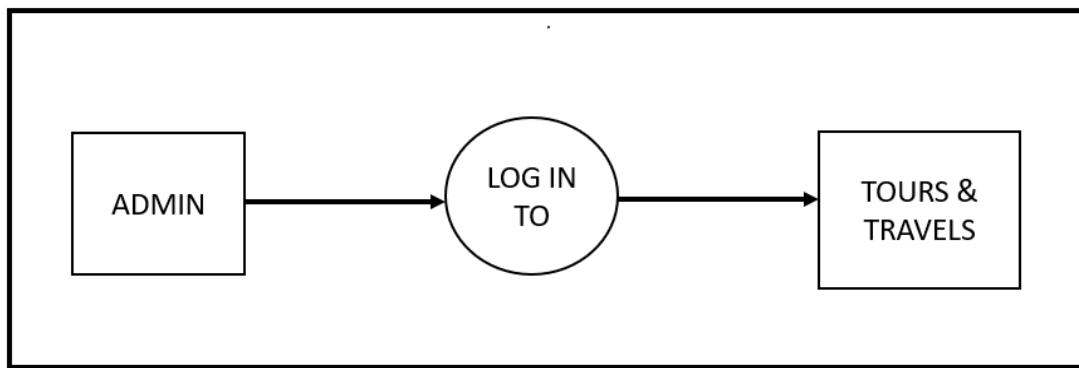
CHAPTER: 6

DATA FLOW DIAGRAM & ER DIAGRAM

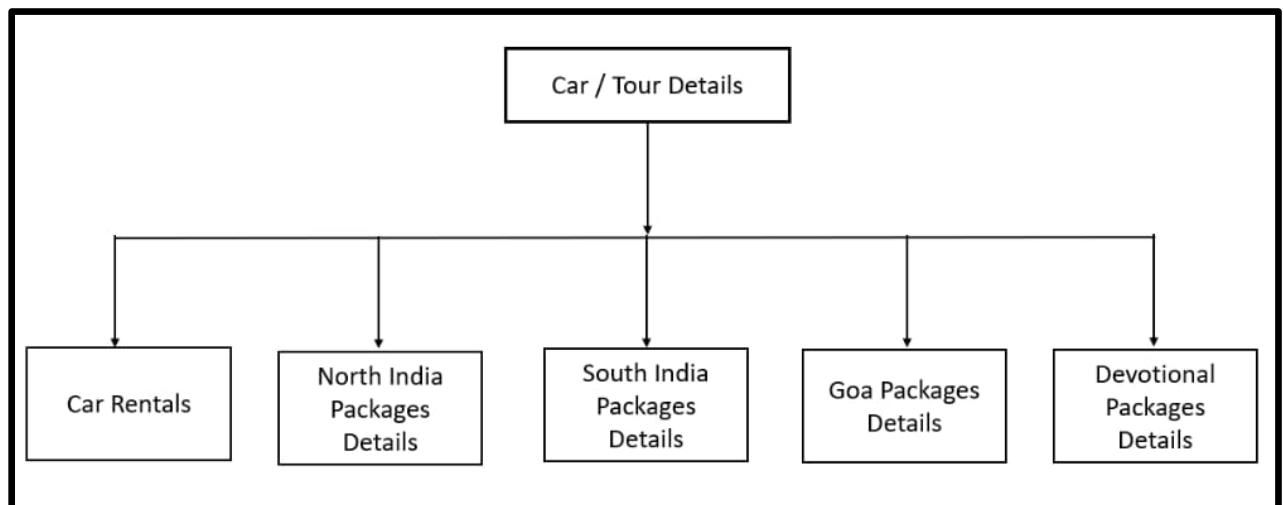
6.1 DATA FLOW ANALYSIS:

The data flow analysis is a study of how and what the data flows through the system. It is done with the help of Data Flow Diagram (DFD), which depicts the entities involved in the system, data input by the entities, processes the input data undergoes and data store where the data is stored.

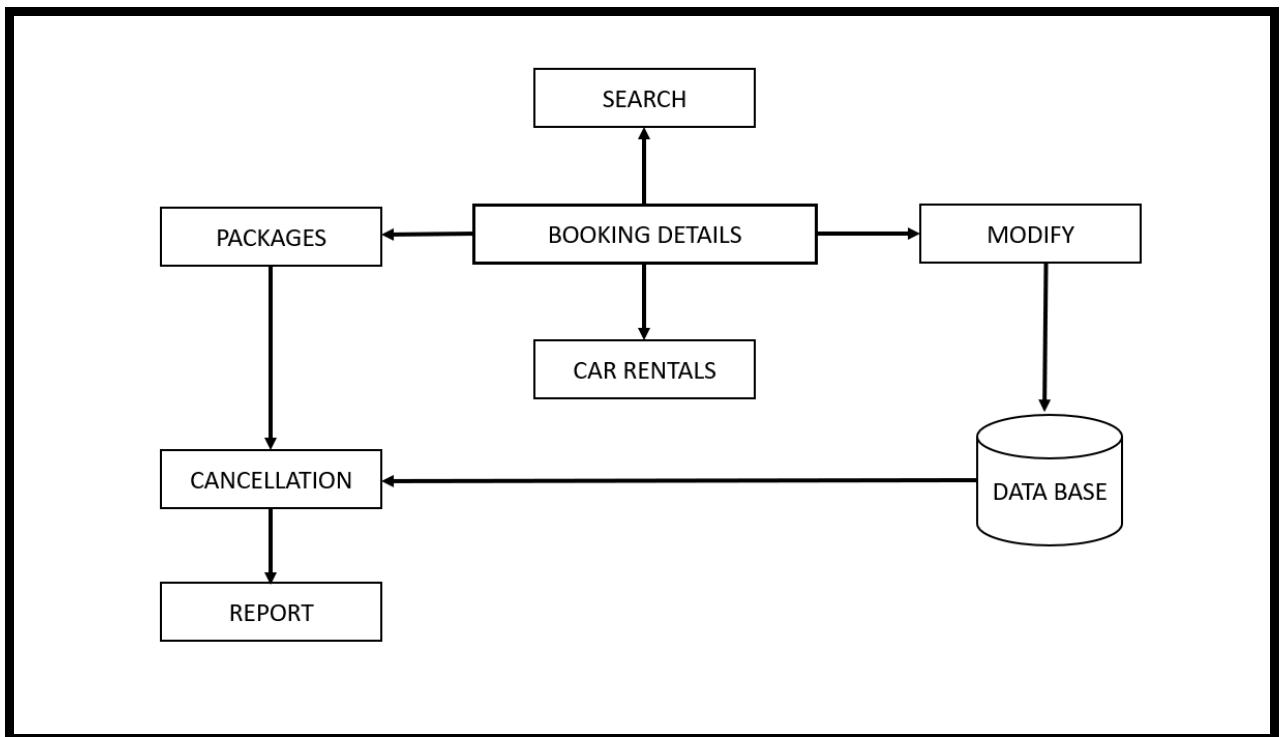
6.1 DFD



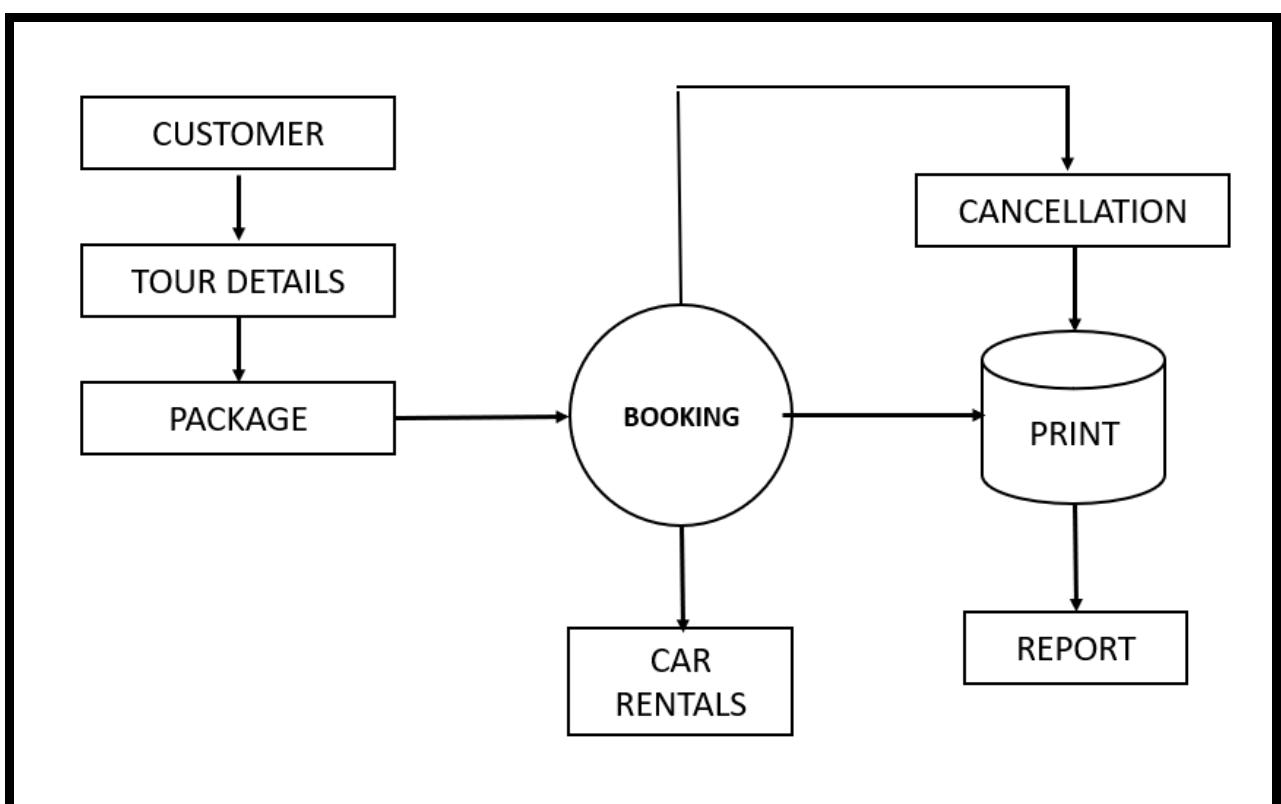
6.2 DFD



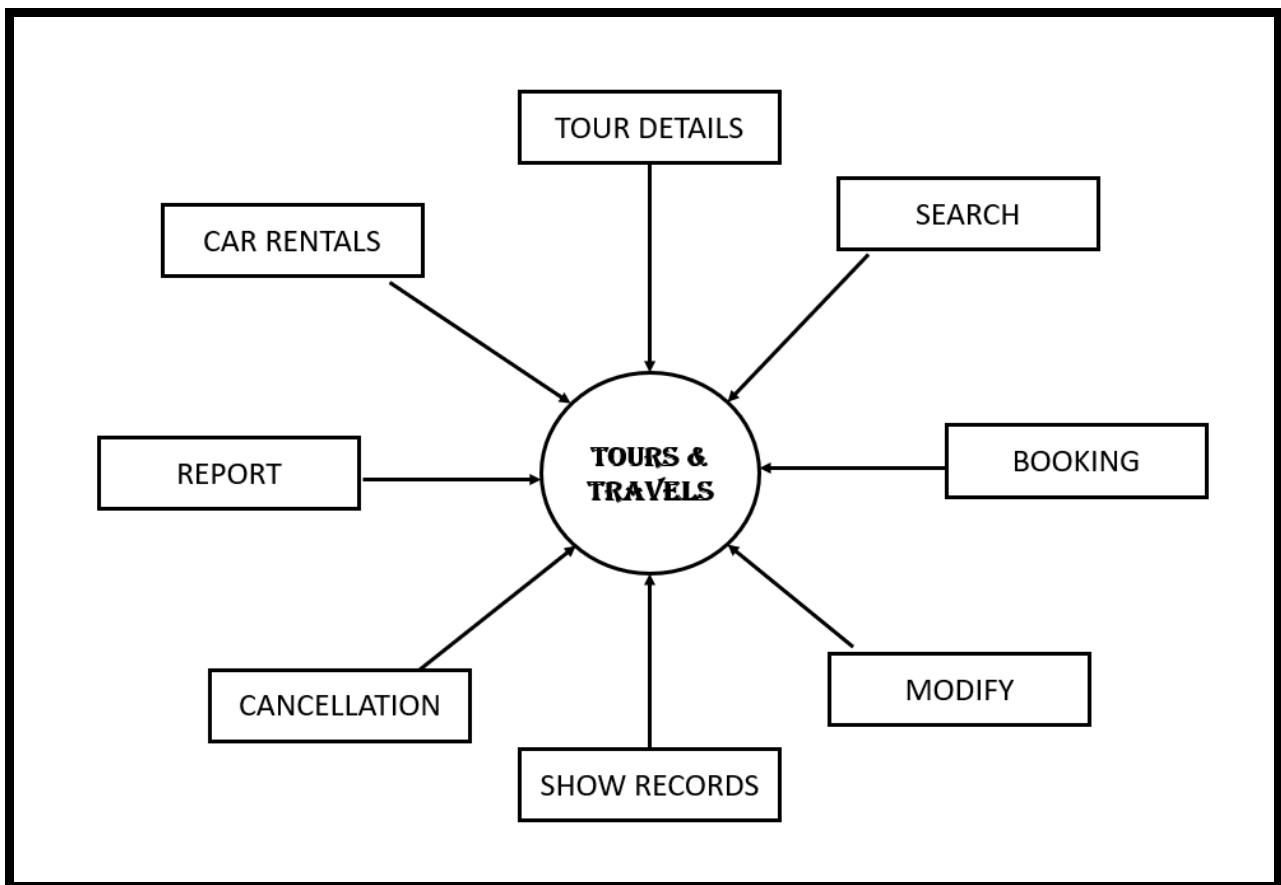
6.3 DFD



6.4 DFD

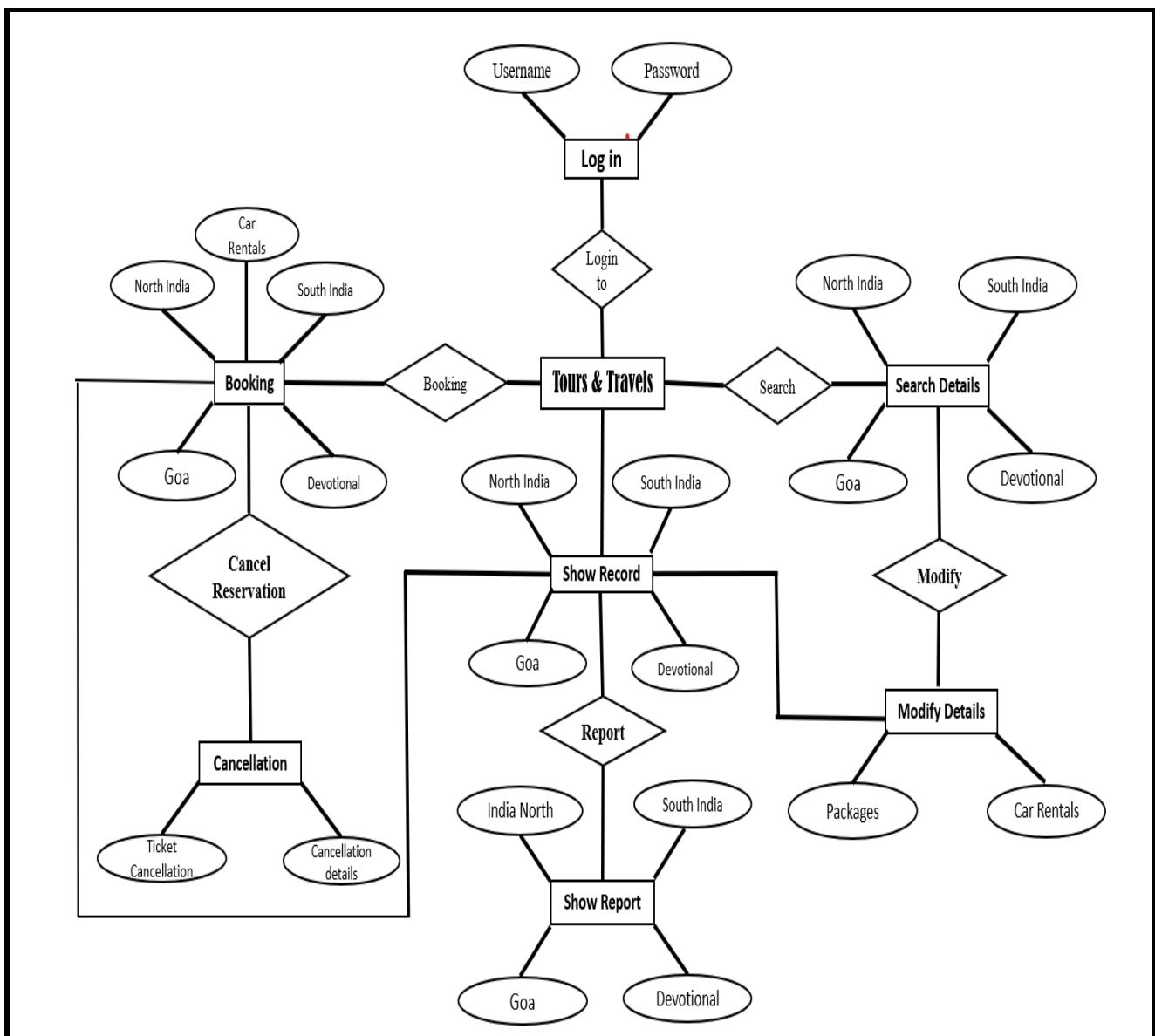


6.5 DFD



6.2 ER DIAGRAM

The entity relationship diagram is used to represent the relationship between one entity database record to another entity. How the project is flowing according to that flow of the project, we are drawing the ER diagram with connectivity from one relation to another as shown below.



CHAPTER NO: 7

INTRODUCTION TO VISUAL BASIC 2010

Microsoft Visual Studio 2010 delivers on the Microsoft vision of smart client applications by enabling developers to rapidly create connected applications that deliver the highest quality, rich user experiences. With Visual Basic 2010, organizations will find it easier than ever before to capture and analyze information to help them make effective business decisions. Visual Basic 2010 enables organizations of every size to rapidly create more secure, manageable, and reliable applications that take advantage of Windows Vista™ and the 2007 Office system.

Visual Basic 2010 delivers key advances for developers in three primary pillars:

Rapid application development

Effective team collaboration

Break through user experiences

Visual Basic 2010 provides advanced development tools, debugging features, database functionality, and innovative features for quickly creating tomorrow's cutting-edge applications across a variety of platforms.

Visual Basic 2010 includes enhancements such as visual designers for faster development with the .NET Framework 3.5, substantial improvements to Web development tools and language enhancements that speed development with all types of data. Visual Basic 2010 provides developers with all the tools and framework support required to create compelling, expressive, AJAX-enabled Web applications.

Developers will be able to take advantage of these rich client-side and server-side, frameworks to easily build client-centric Web applications that integrate with any back-end data provider, run within any modern browser, and have complete access to VB.NET application services and the Microsoft platform.

7.1 RAPID APPLICATION DEVELOPMENT

To help developers rapidly create modern software, Visual Basic 2010 delivers improved language and data features, such as Language Integrated Query (LINQ), that make it easier for individual programmers to build solutions that analyze and act on information.

Visual Basic 2010 also provides developers with the ability to target multiple versions of the .NET Framework from within the same development environment. Developers will be able to build applications that target the .NET Framework 2.0, 3.0 or 3.5, meaning that they can support a wide variety of projects in the same environment.

7.2 BREAK THROUGH USER EXPERIENCE

Visual Basic 2010 offers developer's new tools that speed creation of connected applications on the latest platforms including the Web, Windows Vista, Office 2007, SQL Server 2010 and Windows Server 2008. For the Web, ASP.NET AJAX and other new technologies will enable developers to quickly create a new generation of more efficient, interactive, and personalized Web experiences.

7.3 EFFECTIVE TEAM COLLABORATION

Visual Basic 2010 delivers expanded and improved offerings that help improve collaboration in development teams, including tools that help integrate database professionals and graphic designers into the development process.

Use the Microsoft .NET Framework 3.5

The .NET Framework enables the rapid construction of connected applications that provide outstanding end-user experiences by providing the building blocks (pre-fabricated software) for solving common programming tasks. Connected applications built on the .NET Framework model business processes effectively and facilitate the integration of systems in heterogeneous environments.

Together Visual Studio and the .NET Framework reduce the need for common plumbing code, reducing development time and enabling developers to concentrate on solving business problems.

The .NET Framework 3.5 builds incrementally on the .NET Framework 3.0. Enhancements have been made to feature areas including the base class library, Windows Workflow

Foundation, Windows Communication Foundation, Windows Presentation Foundation, and Windows Card Space.

7.4 OPERATORS

Programming languages have a set of operators that perform arithmetical operations, and others such as Boolean operations on truth values, and string operators manipulating strings of text. Computers are mathematical devices, but compilers and interpreters require a full.

Syntactic theory of all operations in order to parse formulae involving any combinations correctly. In particular, they depend on operator precedence rules, on order of operations, that are tacitly assumed in mathematical writing.

Conventionally, the computing usage of operator also goes beyond the mathematical usage (for functions). In Visual Basic.NET, New, Address of and CType are operators. You can also define your own uses for operators. When an operator is alphanumeric rather than a punctuation character, it is sometimes called a named operator.

So operators are special symbols that are used to represent for example simple computations like addition and multiplication. Most of the operators in VB.NET do exactly what you would expect them to do, because they are common mathematical symbols. For example, the operator for adding two integers is +.

7.5 VISUAL BASIC CONTROLS

Like its predecessor, Visual Basic excels in creating graphical user interfaces. Although the programmer still has the option of creating controls and setting their properties programmatically (i.e., by hand writing the code), he usually will create many, if not all of the controls in the interface by selecting items from the toolbox and adding to a particular form. While working with forms, you can use the toolbox to drag different controls to the form you are designing, resize them and relocate them using the mouse, and set the control's properties in a corresponding properties window to quickly develop the user interface. Events handlers for each control's most common event can be quickly created by double-clicking on the control to create a new event handler and be sending to that event handler in the code window.

7.6 VISUAL BASIC INHERITANCE

Inheritance is mainly used to reduce duplication of code. By using the inherits keyword, you can extend and modify an existing class to have additional properties and methods.

For example, imagine we have an existing class called "Person":

```
Public Class Person
    Public FirstName as String
    Public LastName as String
    Public DateOfBirth as Date
    Public Gender as String
    Public ReadOnly Property FullName () As String
        Get
            Return FirstName & " " & LastName
        End Get
    End Property
End Class
```

Now imagine we wanted to create a special class called "Customer", which had all the properties of "Person", but also additional properties called "CustomerID" and "CustomerType". We could just make a different class with similar properties as follows:

```
Public Class Customer
    Public FirstName as String
    Public LastName as String
    Public DateOfBirth as Date
    Public Gender as String
    Public ReadOnly Property FullName () As String
        Get
            Return FirstName & " " & LastName
        End Get
    End Property
    Public CustomerID as String
    Public CustomerType as String
End Class
```

An alternative approach; however, is to use the "inherits" keyword as follows:

```
Public Class Customer
    Inherits Person
    Public Customer ID as String
    Public Customer Type as String
End Class
```

"Inherits Person" automatically gives the new "Customer" class all the properties and methods of the "Person" class, as well as the two new properties. This approach also has several advantages:

We don't have to re-type the First Name, Last Name, DateOfBirth etc. properties

The Full Name property only needs to be debugged in one place

Any code that uses a "Person" object can also use a "Customer" object.

7.7 VISUAL BASIC CLASSES

The Class concept is the main foundation of Object-Oriented Programming. In a world of Graphical User Interfaces and more complex programs, classes have become a very important part of programming.

7.7.1 Terminology

- **Object:** A unit that has its own properties and methods for a user to use at his or her discussion.
- **Encapsulation:** Lets the user of the class control the data and operations of a class that can be seen from other classes.
- **Property:** Represents a data value associated with an instance.
- **Method:** An operation that can be performed by the class.
- **Constructor:** The method that is called when an object of the class has been instantiated.
- **Field:** A variable at the class level.

7.8 IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. Once the design is complete, most of the major decisions about the system have been made. The goal of coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim in this phase is to implement the design in the best possible manner.

The coding phase affects both testing and maintenance profoundly. Since the testing and maintenance costs of software are much higher than the coding cost, the goal of the coding should be to reduce the testing and maintenance effort. Hence, during coding the focus should be on developing the programs that are easy to read and understand, and not simply on developing programs that are easy to write.

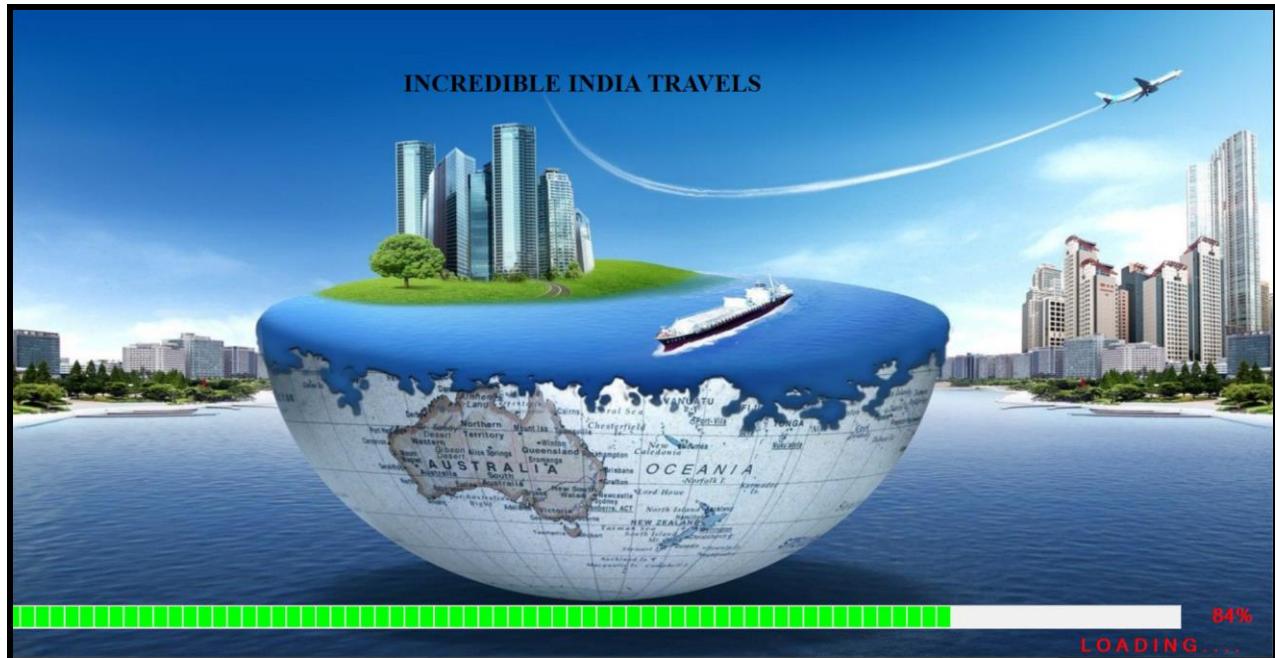
The implementation stage consists of

- Making the necessary changes of the system as desired by the user.
- Training the user personal prior to the implementation of two steps shown below as to be carried out.
- Testing the developed programs with the sample data.
- Detection and correction of errors.

CHAPTER: 8

INPUT/OUTPUT SCREENS

8.1 HOME PAGE



8.2 LOGIN PAGE



8.3 COMPANY DETAILS

COMPANY DETAILS

Name of the Company	INCREDIBLE INDIA TRAVELS
Location and Address	#15/16,1st Cross 5th Main,2nd Block Muddinapalya main road, Nagabhatti,Bangalore-560091
Contact Person	Mr Tejas Mr Shyam
Contact No.	6361054381 / 6360146152
EmailID	shyamn2001@gmail.com tejasr4381@gmail.com

INCREDIBLE INDIA TRAVELS

8.4 RULES AND REGULATIONS

TERMS AND CONDITIONS

-> When you book a package:

1. You have one point of contact who is an agent with intimate knowledge of the destinations you plan to visit.
2. Team members are ready to assist you through out your trip, providing you with specifics in advance of each day.
3. Coordinated itineraries include airport transfer service, accommodations, sightseeing activities, and certain meals for one fixed price.
4. You work with quality, licensed, insured, agency-trusted providers, guides, and drivers.
5. You get peace of mind from having everything included and taken care of for you in one package.

-> When you book your own trip:

1. Contacting and coordinating plans with several, individual providers can expand your network of resources in the destination.
2. Organizing your own schedule and details after researching specifics means learning about rules and guidelines for visitors.
3. Comparing prices and flight schedules can land you really good deals on airfare and hotels so long as you keep trying.
4. Discovering things on your own is a continuous learning experience.
5. Conveying your questions or issues to various providers, guides, or drivers across different language barriers can be an adventure.

8.5 MDI FORM



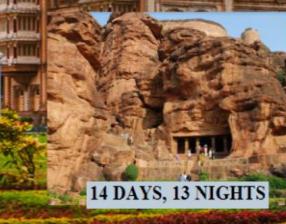
8.6 TOURS DETAILS

Tour Details			
NORTH INDIA PACKAGE	GOA PACKAGE	DEVOTIONAL PACKAGE	SOUTH INDIA PACKAGE
<ul style="list-style-type: none"> - AC accomodation - Veg/Non Veg - Sightseeing tours - Transfers 	<ul style="list-style-type: none"> - AC accomodation - Veg/Non Veg - Sightseeing tours - Transfers 	<ul style="list-style-type: none"> - Non AC accomodation - Veg only - Sightseeing tours - Transfers 	<ul style="list-style-type: none"> - Non AC accomodation - Veg/Non Veg - Sightseeing tours - Transfers
Starting from ₹ 22,000/- (per person)	Starting from ₹ 14,500/- (per person)	Starting from ₹ 15,000/- (per person)	Starting from ₹ 18,500/- (per person)

8.7 NORTH INDIA TOURS DETAILS

 7 DAYS, 6 NIGHTS Kashmir, Ladakh, Manali, Delhi, Mussoorie, Jaipur, Dalhousie, Delhi, Agra, Ladakh, Amritsar	 14 DAYS, 13 NIGHTS Manali, Shimla, Mussoorie, Nainital, Dalhousie, Jaipur, Ladakh, Srinagar, Taj Mahal, Lotus temple, Gulmarg, Red fort, Amritsar, Qutub minar, Kasol, Udaipur, Dehradun, Dharamsala, Jim Corbett National park, Dalhousie.	 21 DAYS, 20 NIGHTS Haridwar, Amritsar, Jaipur, Srinagar, Kasol, Bir, India Gate, Red fort, Taj Mahal, Manali, Varanasi, Dharamshala, Varamasi, Gulmarg, Kasauli, Spiti, Darjeeling, Allahabad, Vaishno Devi, Mussoorie, Ladakh, Shimla, Manali, Darjeeling, Lucknow, Dharamsala, Dalhousie, Nainital, Dehradun.	 25 DAYS, 24 NIGHTS Manali, Mussoorie, Rishikesh, Shimla, Nainital, Dalhousie, Srinagar, Varanasi, Jaipur, Agra, Ladakh, Taj Mahal, Dharamshala, Udaipur, Spiti valley, Himachal Pradesh, Delhi, Amritsar, Gulmarg, Mount Abu, Kasol, Red Fort, Qutub minar, Darjeeling, Amarnath, Kasauli, Chandigarh, Uttarakhand, Kufri, Rajaji National park, Ranikhet, Gangotri, Kinnaur.
Starting from ₹ 22,000/- (per person)	Starting from ₹ 38,500/- (per person)	Starting from ₹ 55,500/- (per person)	Starting from ₹ 68,500/- (per person)

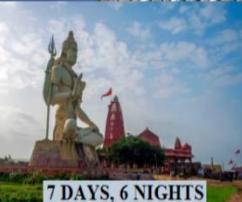
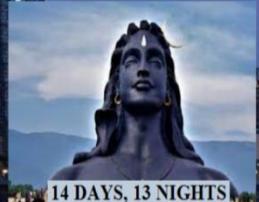
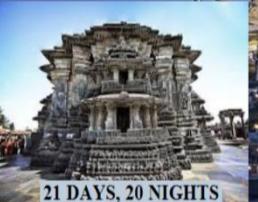
8.8 SOUTH INDIA TOURS DETAILS

 7 DAYS, 6 NIGHTS Coorg, Kodaikanal, Ooty, Gokarna, Hampi, Jog Falls, Kanyakumari, Mysore, Chikmagalur.	 14 DAYS, 13 NIGHTS Hyderabad, Alleppey, Munnar, Chikmagalur, Dandeli, Mysore, Kanyakumari, Coonoor, Gokarna, Kodagu, Rameshwaram, Coimbatore, Gol Gumbaz, Jog falls, Hampi.	 21 DAYS, 20 NIGHTS Mumbai, Gol Gumbaz, Jog falls, Malabar beach, Ooty, Bangalore, Shravanabelagola, Coimbatore, Hampi, Hyderabad, Kochi, Kanyakumari, Araku Valley, Thrissur, Kodagu, Chikmagalur, Varkala, Lepakshi, Ramoji film city, Pondicherry, Andaman and Nicobar, Kochi, Wayanad, Munnar.	 25 DAYS, 24 NIGHTS Bangalore, Coimbatore, Araku Valley, Irivandrum, Kochi, Kanchipuram, Mahabalipuram, Gokarna, Kodaikanal, Coorg, Ooty, Coonoor, Hampi, Tanjore, Pondicherry, Alleppey, Kanyakumari, Hyderabad, Chikmagalur, Rameshwaram, Thrissur, Mysore, Warangal, Munnar, Madurai, Gol Gumbaz, Gateway of India, Elephanta caves, Madh Island, Wayanad, Araku Valley, Pondicherry, Andaman and Nicobar.
Starting from ₹ 18,500/- (per person)	Starting from ₹ 30,500/- (per person)	Starting from ₹ 48,500/- (per person)	Starting from ₹ 59,500/- (per person)

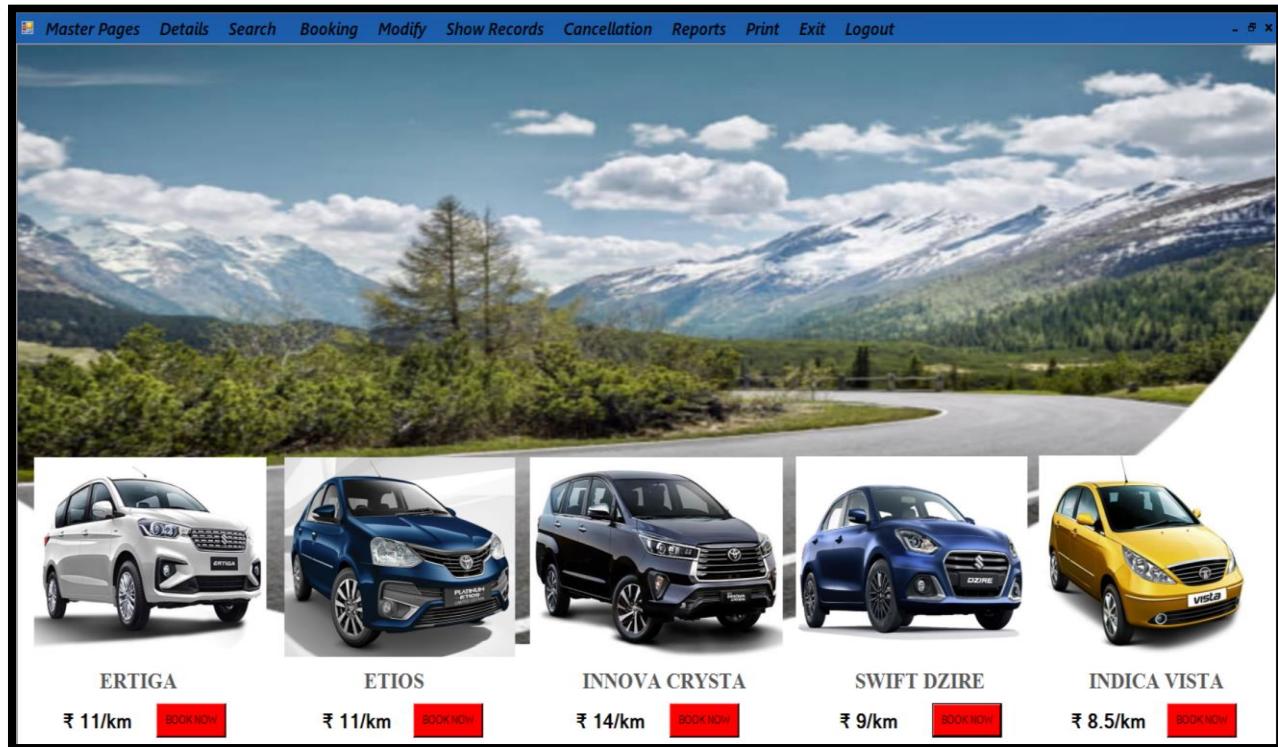
8.9 GOA TOURS DETAILS

 7 DAYS, 6 NIGHTS				 14 DAYS, 13 NIGHTS				 21 DAYS, 20 NIGHTS				 25 DAYS, 24 NIGHTS			
Calangute Beach,Baga Beach,Palolem Beach, Bom Jesus,Aguada Fort,Mangeshi Temple, Chapora Fort,Cruise,Anjuna Beach.				Calangute,Baga,Palolem,Panaji,Basicika,Aguada Fort, Anjuna Beach,SeCathedral,Chapora Fort,Arambol Beach, Candolim Beach,Butterfly Beach,Vagator Beach, ShreeMangeshiTemple,AshvemBeach,Majorda Beach, Vraca Beach.				Calangute Beach,Baga Beach,Palolem Beach, Bom Jesus,Aguada Fort,Mangeshi Temple,Cruise, Dudhsagar falls,Anjuna beach,Chapora fort,Brittis, Candolim beach,Grande Island,Bhagawan mahariv wildlife sanctuary,Arvalem waterfalls,Molem National park.				Calangute Beach,Anjuna Beach,Baga Beach, Panaji,Agoda Fort,Casino,Chapora Fort,Divar Island,Dona Paula,Vagator Beach,Ashwem Beach,Panji,Mahadeva temple,Mangeshi temple,Bambolim beach,Molem National Park, Arvalem waterfalls,Dudhsagar waterfalls,Grande Island,Fort Tiracol,Candolim beach.			
Starting from	₹ 14,500/-	Starting from	₹ 25,500/-	(per person)		Starting from	₹ 38,000/-	(per person)		Starting from	₹ 45,500/-	(per person)			

8.10 DEVOTIONAL TOURS DETAILS

 7 DAYS, 6 NIGHTS		 14 DAYS, 13 NIGHTS		 21 DAYS, 20 NIGHTS		 25 DAYS, 24 NIGHTS							
Dharmastala,Kukke,Homadu,Sringeri,Mantaralaya, Siridi,Tirupathi,Kalahasti,Mantaralaya, Kanchi,Kamakshi,Kanyakumari.		Dharmastala,Kukke,Homadu,Springer,Mantaralaya, Tirupathi,Kalahasti,Shirdi,Kanyakumari,Kanchi, Sigandur,Kolluru.Adichuchanagiri,Ujire,Nanjangudu, MM Hills,Udupi,Srirangapatna,Shravabelaoagola, Aneggudda,Talakaveri,Idagunji,Coimbatore.		Shirsi,Melukote,Biligi Ranga Hills,Kudalasangama, kotilingeshwara,Dharmastala,Kukke,Homadu,Sringeri, Mantaralaya,Tirupathi,Kalahasti,Kanyakumari,Kanchi, Kamakshi,Sigandur,Kolluru,Adichuchanagiri,Ujire, Nanjangudu,MM Hills,Udupi,Srirangapatna,Siridi, Shravabelaoagola,Aneggudda,Talakaveri,Yadiyur.		Thirupathy,Dhamastala,Sringeri,Homadu,Idagunji, Belur,Halebidu,Kukke,Kanyakumari,Kalahasti,Udupi, Sommath temple,Vinipaksha temple,Lotus temple, Madurai,Shirdi,Varanasi,Golden temple,Isckon, Sun temple,Sanchi stupa,Anamata cave temple, Kedamath temple,Yamunotri,Bhadrinath temple, Birla mandir,Gangotri temple,Coimbatore.							
Starting from	₹ 15,000/-	(per person)	Starting from	₹ 28,500/-	(per person)	Starting from	₹ 45,500/-	(per person)	Starting from	₹ 55,500/-	(per person)		

8.11 CAR RENTALS DETAILS



8.12 SEARCH FORM

INCREDIBLE INDIA TRAVELS

24-01-2022


SEARCH PASSENGER BOOKING DETAILS

Select Passenger ID	<input type="text"/>	No. of days	<input type="text"/>
Name	<input type="text"/>	Total	<input type="text"/>
Gender	<input type="text"/>	Places	<input type="text"/>
Contact number	<input type="text"/>		
Email ID	<input type="text"/>		
Address	<input type="text"/> <input type="text"/> <input type="text"/>		
Date of trip	<input type="text" value="24 January 2022"/>		



8.13 NORTH INDIA BOOKING

Master Pages Tour Details Search Booking Modify Show Records Cancellation Reports Print Exit Logout

INcredible!INDIA TRAVELS

Incredible India

24-01-2022

NORTH INDIA PACKAGES

Passenger ID	<input type="text"/>	No. of days	<input type="text"/>
Name	<input type="text"/>	Total	<input type="text"/>
Gender	<input type="text"/>	Places	<input type="text"/>
<input type="radio"/> MALE <input type="radio"/> FEMALE <input type="radio"/> OTHERS			
Contact number	<input type="text"/>		
Email ID	<input type="text"/>		
Address	<input type="text"/> <input type="text"/> <input type="text"/>		
Date of trip	24 January 2022		



8.14 CAR RENTALS

Master Pages Details Search Booking Modify Show Records Cancellation Reports Print Exit Logout

INcredible!INDIA TRAVELS

08-02-2022

CAR RENTAL SERVICES

Passenger ID	<input type="text"/>	Select Vehicle	<input type="text"/>
Name	<input type="text"/>	Total KM	<input type="text"/>
Gender	<input type="text"/>	Driver Fees	<input type="text"/>
<input type="radio"/> MALE <input type="radio"/> FEMALE <input type="radio"/> OTHERS		Amount	<input type="text"/>
Contact number	<input type="text"/>		
Email ID	<input type="text"/>		
Address	<input type="text"/> <input type="text"/> <input type="text"/>		
Date of trip	08 February 2022		



8.15 MODIFY FORM

Master Pages Details Search Booking Modify Show Records Cancellation Reports Print Exit Logout

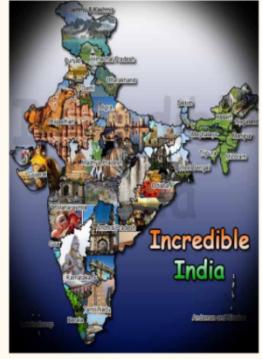
INCREDIBLE !INDIA TRAVELS

08-02-2022



MODIFY PASSENGER BOOKING DETAILS

Select Passenger ID	<input type="text"/>	Date of trip	<input type="text" value="08 February 2022"/>
Name	<input type="text"/>	No. of days	<input type="text"/>
Gender	<input type="text"/>	Total	<input type="text"/>
Contact number	<input type="text"/>	Places	<input type="text"/>
Email ID	<input type="text"/>		
Address	<input type="text"/>		
<input type="radio"/> MALE <input type="radio"/> FEMALE <input type="radio"/> OTHERS			
<input type="button" value="MODIFY"/> <input type="button" value="CANCEL"/> <input type="button" value="PRINT"/> <input type="button" value="EXIT"/>			



PassengerID	Name	Gender	Contact	EmailID	Add1	Add2	Add3	Date	Days	Total	Places
1001	JOHN	MALE	6963012050	john32@gmail.c...	#77	RR LAYOUT	BANGALORE	21 January 2022	21	23000	Manali, Shim...
1002	SCOFIELD	MALE	7795865118	scofield333@gm...	#77	JAYANAGAR	BANGALORE	12 January 2022	30	27200	Calangute, ...
1003	SARA	FEMALE	6368740125	sara077@yahoo...	#56,	RR Nagar	Bangalore	18 February 2022	7	5500	Kashmir, Lad...
1004	PRAJWAL	MALE	8475445582	prajwal90@gmail...	#23, 5TH CROSS	MALATHAHALLI	BANGALORE	24 February 2022	25	46500	Markambal...

8.16 CAR RENTALS MODIFY

Master Pages Details Search Booking Modify Show Records Cancellation Reports Print Exit Logout

INCREDIBLE !INDIA TRAVELS

08-02-2022



CAR RENTAL SERVICES

Select Passenger ID	<input type="text"/>	Date of trip	<input type="text" value="08 February 2022"/>
Name	<input type="text"/>	Select Vehicle	<input type="text"/>
Gender	<input type="text"/>	Total KM	<input type="text"/>
Contact number	<input type="text"/>	Driver Fees	<input type="text"/>
Email ID	<input type="text"/>	Amount	<input type="text"/>
Address	<input type="text"/>		
<input type="button" value="MODIFY"/> <input type="button" value="CANCEL"/> <input type="button" value="PRINT"/> <input type="button" value="HOME"/>			



PassengerID	Name	Gender	Contact	EmailID	Add1	Add2	Add3	Date	Vehicle	Totalkm	Driverfees	Amount
101	TEJAS	MALE	6369412512	teja2001@g...	#55,	PAPAREDD...	BANGALORE	11 January ...	ERTIGA	500	500	5500
102	RAVI	MALE	9854711002	sondi@gmail...	#469	S L LAYOUT	BANGALORE	29 January ...	INNOVA	1200	2000	16800
103	ASHWIN	MALE	9663584410	ash67@gm...	#89	ITI LAYOUT	BANGALORE	05 March 20...	ERTIGA	560	1000	6160
104	SHYAM	MALE	6360145253	shyamn98@...	#21	SIR M.V LA...	BANGALORE	26 February...	INNOVA CR...	65	200	910
105	ANKITHA	FEMALE	8952013235	ankithaa34...	#57	RR NAGAR	BANGALORE	19 March 20...	INDICA VISTA	580	200	493

8.17 NORTH INDIA BOOKING DETAILS REPORT

Master Pages Details Search Booking Modify Show Records Cancellation Reports Print Exit Logout

INCREDIBLE !INDIA TRAVELS



NORTH INDIA BOOKING DETAILS

PassengerID	Name	Gender	Contact	EmailID	Add1	Add2	Add3	Date	Days	Total	Places	Package
1001	JOHN	MALE	6963012050	john32@gmail.c...	#77	RR LAYOUT	BANGALORE	21 January 2022	21	23000	Manali, Shimla, Va...	NORTH INDIA P...
1003	SARA	FEMALE	6368740125	sara077@yahoo...	#56	RR Nagar	Bangalore	18 February 2022	7	5500	Kashmir, Ladakh...	NORTH INDIA P...
1008	PAVAN	MALE	9668741002	pavan123@gmail...	#21	VIJAYNAGAR	BANGALORE	05 March 2022	14	16500	Jajpur, Dalhousie...	NORTH INDIA P...
1015	HELSINKI	MALE	8596214100	hel34@gmail.com	#56	RPC LAYOUT	BANGALORE	24 February 2022	30	39000	Manali, Mussoorie...	NORTH INDIA P...
1016	LINCON	MALE	8752200123	linc45@gmail.com	#444	KOTTIGEPELTYA	BANGALORE	11 March 2022	7	5500	Kashmir, Ladakh...	NORTH INDIA P...

[HOMEPAGE](#)

8.18 GOA BOOKING DETAILS REPORT

Master Pages Details Search Booking Modify Show Records Cancellation Reports Print Exit Logout

INCREDIBLE !INDIA TRAVELS



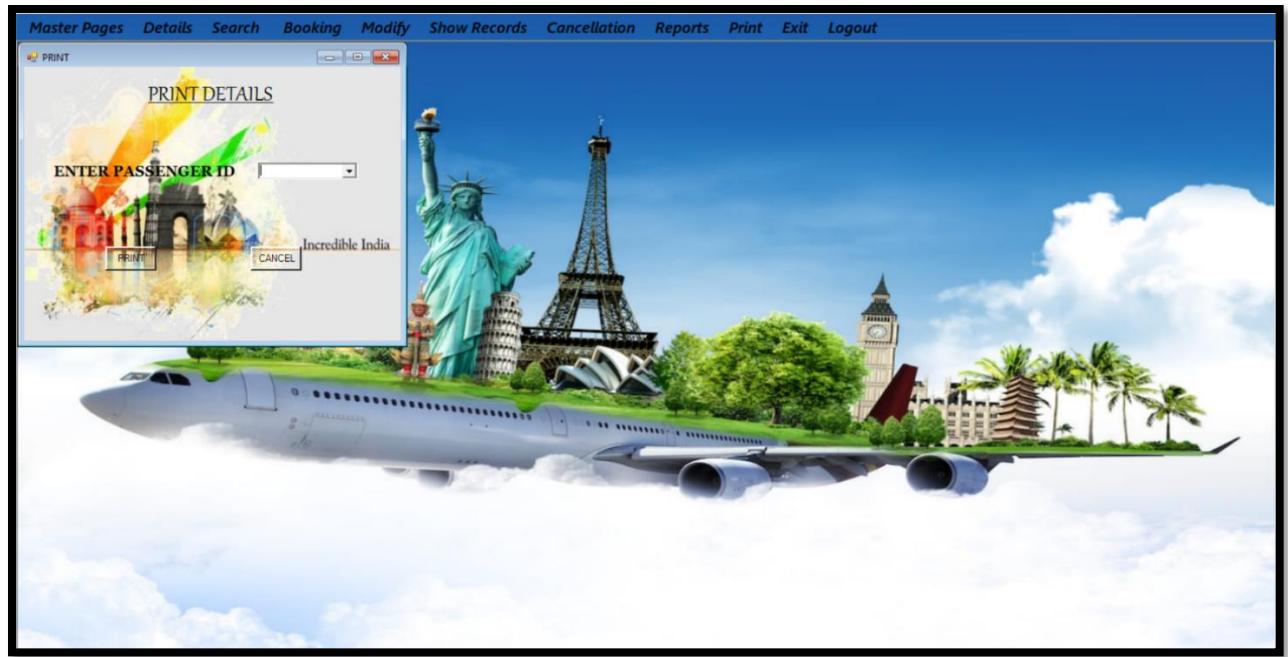
08-02-2022

GOA BOOKING DETAILS

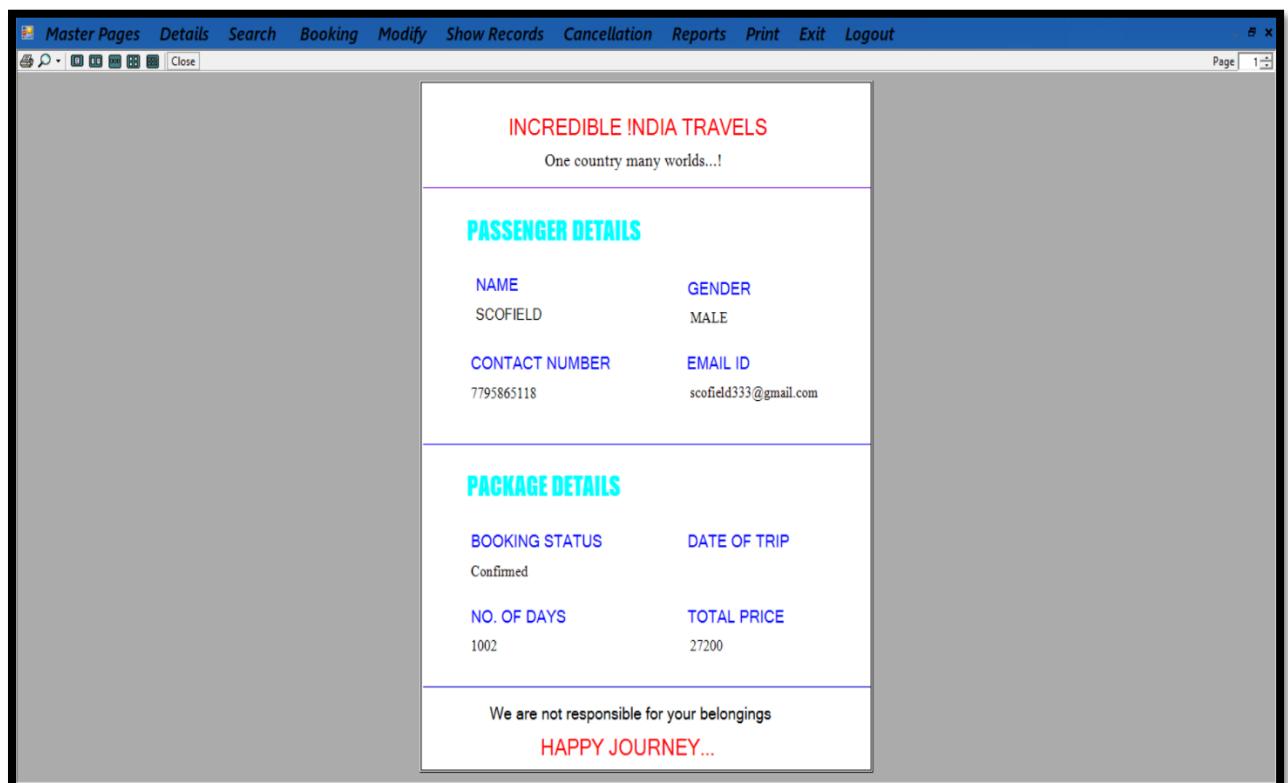
PassengerID	Name	Gender	Contact	EmailID	Add1	Add2	Add3	Date	Days	Total	Places	Package
1002	SCOFIELD	MALE	7795865118	scofield333@gm...	#77	JAYANAGAR	BANGALORE	12 January 2022	30	27200	Calangute, Baga...	GOA PACKAGES
1009	SHYAM	MALE	6360146152	shyamr2001@g...	#3	NAGADEVENAH...	BANGALORE	16 February 2022	25	22500	Calangute, Baga...	GOA PACKAGES
1010	RAJ	MALE	9663595778	raj89@gmail.com	#56	RR LAYOUT	BANGALORE	25 February 2022	14	15000	Calangute Beach...	GOA PACKAGES
1013	DEEPAK	MALE	9885547555	deepu23@gmail...	#56	MALGALLA	BANGALORE	08 February 2022	21	20000	Calangute Beach...	GOA PACKAGES
1014	VIRAT	MALE	7852220012	virat67@gmail.c...	#09	KENGERI	BANGALORE	03 March 2022	14	15000	Calangute Beach...	GOA PACKAGES

[HOMEPAGE](#)

8.19 PRINT TICKET



8.20 PASSENGER TICKET



CHAPTER 9

SAMPLE CODING

HOME PAGE CODING:

```
Public Class homepage
```

```
Dim x As Integer
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Timer1.Tick
```

```
If x < ProgressBar1.Maximum() Then
```

```
ProgressBar1.Value = x
```

```
x = x + 4
```

```
Label2.Text = x & "%"
```

```
Else
```

```
Me.Hide()
```

```
loginpage.Show()
```

```
End If
```

```
End Sub
```

LOGIN PAGE CODING:

```
Imports System.Data.SqlClient
```

```
Public Class loginpage
```

```
Dim temp As VariantType
```

```
Private Sub PictureBox3_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles PictureBox3.Click
```

```
temp = MsgBox("Do you want to exit!(YES/NO)", MsgBoxStyle.YesNo +
```

```
MsgBoxStyle.Exclamation, "exit")
```

```
If temp = vbYes Then
```

```
MsgBox("THANK YOU", MsgBoxStyle.Information, " Best Wishes")
```

```
End
```

```
End If
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button1.Click
```

```
If conn.State = ConnectionState.Open Then conn.Close()
```

```
conn.Open()
```

```
Dim Cmd0 As New SqlCommand("select *from LoginTab where Username=" &
```

```
UCase(TextBox1.Text) & "" and Password=" & TextBox2.Text & "", conn)
```

```
Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
```

```
If D1.HasRows Then
```

```
TTDBMDI.Show()
```

```
Else
```

```
MsgBox("INVALID USERNAME OR PASSWORD", MsgBoxStyle.Critical, "login checkup")
```

```
lable4.Text = ""
```

```
TextBox2.Text = "
```

```
lable4.Focus()
```

```
End If
```

```
End Sub
```

```
End Class
```

TOUR DETAILS FORM CODING:

```
Public Class Tour_details
```

```
Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Label1.Click
```

```
North_India_Details.Show()
```

```
End Sub
```

```
Private Sub Tour_details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles MyBase.Load
```

```
Me.Size = TTDBMDI.Size
```

```
End Sub
```

```
Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles PictureBox2.Click
```

```
North_India_Details.Show()
```

```
End Sub
```

```
Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles PictureBox3.Click
```

```
Goa_Details.Show()
```

```
End Sub
```

```
Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles PictureBox4.Click
```

```
Devotional.Show()
```

```
End Sub
```

```
Private Sub PictureBox5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles PictureBox5.Click
```

```
South_India_details.Show()
```

```
End Sub
```

```
Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Timer4.Tick
```

```
Static count As Integer
```

```
count = count + 1
```

```
Select Case count
```

```
Case 1
```

```
Pics1.BringToFront()
```

```
Case 2
```

```
Pics2.BringToFront()
```

```
Case 3
```

```
Pics3.BringToFront()
```

```
Case 4
```

```
Pics4.BringToFront()
```

```
Case 5
```

```
Pics5.BringToFront()
```

```
End Select
```

```
Select Case count
```

```

Case 1
Pict1.BringToFront()
Case 2
Pict2.BringToFront()
Case 3
Pict3.BringToFront()
Case 4
Pict4.BringToFront()
Case 5
Pict5.BringToFront()
End Select

```

```

Select Case count
Case 1
Pic1.BringToFront()
Case 2
Pic2.BringToFront()

```

```

Case 3
Pic3.BringToFront()
Case 4
Pic4.BringToFront()
Case 5
Pic5.BringToFront()
End Select

```

```

Select Case count
Case 1
Pi1.BringToFront()
Case 2
Pi2.BringToFront()
Case 3
Pi3.BringToFront()
Case 4
Pi4.BringToFront()
Case 5
Pi5.BringToFront()
End Select
If count = 5 Then
count = 0
End If
End Sub

```

```

Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label9.Click
North_India_Details.Show()
End Sub

```

```

Private Sub Label18_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label17.Click
Goa_Details.Show()

```

```
End Sub
```

```
Private Sub Label22_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label3.Click
    Devotional.Show()
End Sub
```

```
Private Sub Label23_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label4.Click
    South_India_Details.Show()
End Sub
```

CAR RENTALS DETAILS CODING:

```
Public Class Car_Details
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Car_Rentals.Show()
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    Car_Rentals.Show()
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    Car_Rentals.Show()
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    Car_Rentals.Show()
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    Car_Rentals.Show()
End Sub
```

```
Private Sub Car_Details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Me.Size = TTDBMDI.Size
End Sub
End Class
```

SEARCH FORM CODING:

```
Imports System.Data.SqlClient
Imports System.Text.RegularExpressions
Public Class SearchForm
```

```

Dim pkvar As String
Sub disRecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from Bookings where PassengerID="" & pkvar &
"", conn)
adp.Fill(DS1)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

Private Sub Searchsi_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Me.Size = TTDBMDI.Size
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select PassengerID from Bookings order by PassengerID",
conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader
While d1.Read
ComboBox1.Items.Add(d1(0).ToString)
End While
disRecords()
TextBox11.Text = Format(Today(), "dd/MM/yyyy")
TextBox2.Enabled = False
TextBox3.Enabled = False
TextBox4.Enabled = False
TextBox5.Enabled = False
TextBox6.Enabled = False
TextBox7.Enabled = False
TextBox8.Enabled = False
TextBox9.Enabled = False
TextBox10.Enabled = False
ComboBox2.Enabled = False
DateTimePicker2.Enabled = False
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select * from Bookings where PassengerID="" &
ComboBox1.Text & "", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
ComboBox1.Text = d1(0).ToString
TextBox2.Text = d1(1).ToString
TextBox3.Text = d1(2).ToString
TextBox4.Text = d1(3).ToString
TextBox5.Text = d1(4).ToString
TextBox6.Text = d1(5).ToString

```

```

TextBox7.Text = d1(6).ToString
TextBox8.Text = d1(7).ToString
DateTimePicker2.Text = d1(8).ToString
ComboBox2.Text = d1(9).ToString
TextBox9.Text = d1(10).ToString
TextBox10.Text = d1(11).ToString
End If
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Me.Close()
End Sub
End Class

```

NORTH INDIA PACKAGES CODING:

```

Imports System.Data.SqlClient
Imports System.Text.RegularExpressions
Public Class North_India_Packages

Dim pkvar As String
Sub saveRecord()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
q1var = "Insert Bookings("
q2var = "values("
q1var = q1var & "PassengerID" & ","
q2var = q2var & "" & TextBox1.Text & ","
q1var = q1var & "Name" & ","
q2var = q2var & "" & UCASE(TextBox2.Text) & ","
q1var = q1var & "Gender" & ","
q2var = q2var & "" & TextBox3.Text & ","
q1var = q1var & "Contact" & ","
q2var = q2var & "" & Val(TextBox4.Text) & ","
q1var = q1var & "EmailID" & ","
q2var = q2var & "" & TextBox5.Text & ","
q1var = q1var & "Add1" & ","
q2var = q2var & "" & TextBox6.Text & ","
q1var = q1var & "Add2" & ","
q2var = q2var & "" & TextBox7.Text & ","
q1var = q1var & "Add3" & ","
q2var = q2var & "" & TextBox8.Text & ","
q1var = q1var & "Date" & ","
q2var = q2var & "" & DateTimePicker2.Text & ","
q1var = q1var & "Days" & ","
q2var = q2var & "" & ComboBox1.Text & ","
q1var = q1var & "Total" & ","
q2var = q2var & "" & TextBox9.Text & ","
q1var = q1var & "Places" & ","
q2var = q2var & "" & TextBox10.Text & ","
q1var = q1var & "Package" & ")"
q2var = q2var & "" & Label1.Text & ")"

```

```

MsgBox("YOUR TICKET HAS BEEN BOOKED SUCCESSFULLY")
Dim cmd1 As New SqlCommand(q1var & q2var, conn)
cmd1.ExecuteNonQuery()
disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox5.Text = ""
TextBox4.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox6.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox1.Focus()
End Sub
Sub direcords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from Bookings", conn)
adp.Fill(DS1)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

Private Sub North_India_Packages_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Me.Size = TTDBMDI.Size
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select Days from Northindia order by Days", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
While d1.Read
ComboBox1.Items.Add(d1(0).ToString)
End While
TextBox11.Text = Format(Today(), "dd/MM/yyyy")
End Sub

Private Sub Butsave_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
If TextBox4.Text.Length < 10 Then
MsgBox("Invalid number")
Else
MsgBox("CONFIRM YOUR TICKET?", MsgBoxStyle.YesNo, "Modify Record")
saveRecord()
End If
End Sub

```

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox3.Text = "MALE"
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox3.Text = "FEMALE"
End Sub
```

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox3.Text = "OTHERS"
End Sub
```

```
Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox3.Enabled = False
End Sub
```

```
Private Sub TextBox11_TextChanged_1(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox11.Text = Format(Today(), "dd/MM/yyyy")
End Sub
```

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    If Not Regex.Match(TextBox2.Text, "^[A-z]*$", RegexOptions.None).Success Then
        MsgBox("PLEASE ENTER THE ALPHABETICAL CHARACERS ONLY!")
        TextBox2.Focus()
        TextBox2.Clear()
    End If
End Sub
```

```
Private Sub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    If Not Regex.Match(TextBox4.Text, "^[0-9]*$", RegexOptions.None).Success Then
        MsgBox("PLEASE ENTER THE DIGITS ONLY!")
        TextBox2.Focus()
        TextBox2.Clear()
    End If
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox1.Enabled = False
End Sub
```

```
Private Sub Butnew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    TextBox1.Text = ""
    TextBox2.Text = ""
```

```

TextBox5.Text = ""
TextBox4.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox6.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox1.Focus()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("Select max(PassengerID) from Bookings", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()

If d1.HasRows Then
d1.Read()
TextBox1.Text = Val(IIf(IsDBNull(d1(0)), 1000, d1(0))) + 1
Else
TextBox1.Text = "1001"
End If
TextBox1.Enabled = False
Butnew.Enabled = True
Butsave.Enabled = True
End Sub

Private Sub Butexit_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Butexit.Click
End
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select Total,Places from Northindia where Days=''" &
ComboBox1.Text & "", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
TextBox9.Text = d1(0).ToString
TextBox10.Text = d1(1).ToString
End If
End Sub

Private Sub ButPRINT_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButPRINT.Click
PRINT.Show()
End Sub
End Class

```

GOA PACKAGES CODING:

```

Imports System.Data.SqlClient
Imports System.Text.RegularExpressions
Public Class Goa_Packages

Dim pkvar As String
Sub saveRecord()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
q1var = "Insert Bookings("
q2var = "values("
q1var = q1var & "PassengerID" & ","
q2var = q2var & "" & TextBox1.Text & ","
q1var = q1var & "Name" & ","
q2var = q2var & "" & UCASE(TextBox2.Text) & ","
q1var = q1var & "Gender" & ","
q2var = q2var & "" & TextBox3.Text & ","
q1var = q1var & "Contact" & ","
q2var = q2var & "" & Val(TextBox4.Text) & ","
q1var = q1var & "EmailID" & ","
q2var = q2var & "" & TextBox5.Text & ","
q1var = q1var & "Add1" & ","
q2var = q2var & "" & TextBox6.Text & ","
q1var = q1var & "Add2" & ","
q2var = q2var & "" & TextBox7.Text & ","
q1var = q1var & "Add3" & ","
q2var = q2var & "" & TextBox8.Text & ","
q1var = q1var & "Date" & ","
q2var = q2var & "" & DateTimePicker2.Text & ","
q1var = q1var & "Days" & ","
q2var = q2var & "" & ComboBox1.Text & ","
q1var = q1var & "Total" & ","
q2var = q2var & "" & TextBox9.Text & ","
q1var = q1var & "Places" & ","
q2var = q2var & "" & TextBox10.Text & ","
q1var = q1var & "Package" & ")"
q2var = q2var & "" & Label1.Text & ")"
MsgBox("YOUR TICKET HAS BEEN BOOKED SUCCESSFULLY")
Dim cmd1 As New SqlCommand(q1var & q2var, conn)
cmd1.ExecuteNonQuery()
disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox5.Text = ""
TextBox4.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox3.Text = ""

```

```

TextBox4.Text = ""
TextBox6.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox1.Focus()
End Sub

Sub disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from Bookings", conn)
adp.Fill(DS1)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

Private Sub ComboBox3_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select Total,Places from Goa where Days='"
& ComboBox1.Text & "'", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
TextBox9.Text = d1(0).ToString()
TextBox10.Text = d1(1).ToString()
End If
End Sub

Private Sub Goa_Packages_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Me.Size = TTDBMDI.Size
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select Days from Goa order by Days", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
While d1.Read
ComboBox1.Items.Add(d1(0).ToString())
End While
TextBox11.Text = Format(Today(), "dd/MM/yyyy")
End Sub

Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
TextBox3.Text = "MALE"
End Sub

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)

```

```
TextBox3.Text = "FEMALE"
End Sub
```

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
TextBox3.Text = "OTHERS"
End Sub
```

```
Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
TextBox3.Enabled = False
End Sub
```

```
Private Sub Butexit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Me.Close()
End Sub
```

```
Private Sub TextBox11_TextChanged_1(ByVal sender As System.Object, ByVal e As
System.EventArgs)
TextBox11.Text = Format(Today(), "dd/MM/yyyy")
End Sub
```

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
If Not Regex.Match(TextBox2.Text, "^[A-z]*$", RegexOptions.None).Success Then
MsgBox("PLEASE ENTER THE ALPHABETICAL CHARACERS ONLY!")
TextBox2.Focus()
TextBox2.Clear()
End If
End Sub
```

```
Private Sub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
If Not Regex.Match(TextBox4.Text, "^[0-9]*$", RegexOptions.None).Success Then
MsgBox("PLEASE ENTER THE DIGITS ONLY!")
TextBox2.Focus()
TextBox2.Clear()
End If
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
TextBox1.Enabled = False
End Sub
```

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
```

```

Dim cmd0 As New SqlCommand("select TotalPlaces from Goa where Days="" &
ComboBox1.Text & "", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
TextBox9.Text = d1(0).ToString
TextBox10.Text = d1(1).ToString
End If
End Sub
Private Sub ButNew_Click_2(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButNew.Click
TextBox1.Text = ""
TextBox2.Text = ""
TextBox5.Text = ""
TextBox4.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox6.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox1.Focus()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("Select max(PassengerID) from Bookings", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
TextBox1.Text = Val(IIf(IsDBNull(d1(0)), 2000, d1(0))) + 1

Else
TextBox1.Text = "2001"
End If
TextBox1.Enabled = False
ButNew.Enabled = True
Butsave.Enabled = True
End Sub

Private Sub Butsave_Click_3(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Butsave.Click
If TextBox4.Text.Length < 10 Then
MsgBox("Invalid number")
Else
MsgBox("CONFIRM YOUR TICKET?", MsgBoxStyle.YesNo, "Modify Record")
saveRecord()
End If
End Sub

```

```

Private Sub ButEXIT_Click_2(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButEXIT.Click
End
End Sub

Private Sub ButPRINT_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButPRINT.Click
PRINT.Show()
End Sub
End Class

```

MODIFY CODING:

```

Imports System.Data.SqlClient
Imports System.Text.RegularExpressions
Imports System.Drawing.Size
Public Class Modify

Dim pkvar As String
Sub saveRecord()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
q1var = "Insert Bookings("
q2var = "values("
q1var = q1var & "PassengerID" & ","
q2var = q2var & "" & TextBox1.Text & ","
q1var = q1var & "Name" & ","
q2var = q2var & "" & UCASE(TextBox2.Text) & ","
q1var = q1var & "Gender" & ","
q2var = q2var & "" & TextBox3.Text & ","
q1var = q1var & "Contact" & ","
q2var = q2var & "" & Val(TextBox4.Text) & ","
q1var = q1var & "EmailID" & ","
q2var = q2var & "" & TextBox5.Text & ","
q1var = q1var & "Add1" & ","
q2var = q2var & "" & TextBox6.Text & ","
q1var = q1var & "Add2" & ","
q2var = q2var & "" & TextBox7.Text & ","
q1var = q1var & "Add3" & ","
q2var = q2var & "" & TextBox8.Text & ","
q1var = q1var & "Date" & ","
q2var = q2var & "" & DateTimePicker2.Text & ","
q1var = q1var & "Days" & ","
q2var = q2var & "" & ComboBox1.Text & ","
q1var = q1var & "Total" & ","
q2var = q2var & "" & TextBox9.Text & ","
q1var = q1var & "Places" & ","
q2var = q2var & "" & TextBox10.Text & ","
q1var = q1var & "Package" & ")"
q2var = q2var & "" & Label1.Text & ")"
MsgBox("Data has been save successfully")

```

```

Dim cmd1 As New SqlCommand(q1var & q2var, conn)
cmd1.ExecuteNonQuery()
disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox5.Text = ""
TextBox4.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox1.Focus()
End Sub

```

```

Sub disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from Bookings", conn)
adp.Fill(DS1)
DG1.DataSource = DS1.Tables(0)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

```

```

Private Sub Butdelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Butdelete.Click
If vbNo = MsgBox("ARE YOU SURE YOU WANT TO MODIFY THIS RECORD",
MsgBoxStyle.YesNo, " Delete") Then Exit Sub
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd1 As New SqlCommand("Delete from Bookings where PassengerID=" &
Val(TextBox1.Text) & "", conn)
cmd1.ExecuteNonQuery()
If conn.State = ConnectionState.Open Then conn.Close()
disrecords()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
DateTimePicker2.Text = ""

```

```
TextBox1.Focus()
```

```
End Sub
```

```
Private Sub DG1_CellContentClick_1(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick
    pkvar = DG1.CurrentRow.Cells(0).Value
    If conn.State = ConnectionState.Open Then conn.Close()
    conn.Open()
    Dim cmd0 As New SqlCommand("select * from Bookings where PassengerID=" & pkvar & "", conn)
    Dim d1 As SqlDataReader = cmd0.ExecuteReader()
    If d1.HasRows Then
        d1.Read()
        TextBox1.Text = d1(0).ToString()
        TextBox2.Text = d1(1).ToString()
        TextBox3.Text = d1(2).ToString()
        TextBox4.Text = d1(3).ToString()
        TextBox5.Text = d1(4).ToString()
        TextBox6.Text = d1(5).ToString()
        TextBox7.Text = d1(6).ToString()
        TextBox8.Text = d1(7).ToString()
        DateTimePicker2.Text = d1(8).ToString()
        ComboBox1.Text = d1(9).ToString()
        TextBox9.Text = d1(10).ToString()
        TextBox10.Text = d1(11).ToString()
        Label1.Text = d1(12).ToString()
    Else
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
        TextBox9.Text = ""
        DateTimePicker2.Text = ""
        ComboBox1.Text = ""
        TextBox10.Text = ""
    End If
End Sub
```

```
Private Sub Butsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    saveRecord()
End Sub
```

```
Private Sub Butexit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Butexit.Click
End
End Sub
```

```

Private Sub ModifyDV_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Me.Size = TTDBMDI.Size
TextBox12.Text = Format(Today(), "dd/MM/yyyy")
disrecords()
End Sub

```

```

Private Sub Butmodify_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Butmodify.Click
If vbNo = MsgBox("ARE YOU SURE YOU WANT TO MODIFY THIS RECORD",
MsgBoxStyle.YesNo, "Modify Record") Then Exit Sub
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd1 As New SqlCommand("Delete from Bookings where PassengerID=" &
Val(TextBox1.Text) & "", conn)
cmd1.ExecuteNonQuery()
If conn.State = ConnectionState.Open Then conn.Close()
saveRecord()
End Sub

```

```

Private Sub ButPRINT_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButPRINT.Click
PP1.Show()
End Sub
End Class

```

CAR RENTALS CODING:

```

Imports System.Data.SqlClient
Public Class Car_Rentals

Dim pkvar As String
Sub saveRecord()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
q1var = "Insert into CarRentals("
q2var = "values("
q1var = q1var & "PassengerID" & ","
q2var = q2var & "" & Val(TextBox1.Text) & ","
q1var = q1var & "Name" & ","
q2var = q2var & "" & UCCase(TextBox2.Text) & ","
q1var = q1var & "Gender" & ","
q2var = q2var & "" & TextBox3.Text & ","
q1var = q1var & "Contact" & ","
q2var = q2var & "" & TextBox4.Text & ","
q1var = q1var & "EmailID" & ","
q2var = q2var & "" & TextBox5.Text & ","
q1var = q1var & "Add1" & ","
q2var = q2var & "" & TextBox6.Text & ","
q1var = q1var & "Add2" & ","
q2var = q2var & "" & TextBox7.Text & ","

```

```

q1var = q1var & "Add3" & ","
q2var = q2var & "" & TextBox8.Text & ","
q1var = q1var & "Date" & ","
q2var = q2var & "" & DateTimePicker2.Text & ","
q1var = q1var & "Vehicle" & ","
q2var = q2var & "" & ComboBox1.Text & ","
q1var = q1var & "Totalkm" & ","
q2var = q2var & "" & Val(TextBox9.Text) & ","
q1var = q1var & "Driverfees" & ","
q2var = q2var & "" & Val(TextBox10.Text) & ","
q1var = q1var & "Amount" & ")"
q2var = q2var & "" & Val(TextBox11.Text) & ")"
MsgBox("Data has been save successfully")
Dim cmd1 As New SqlCommand(q1var & q2var, conn)
cmd1.ExecuteNonQuery()
If conn.State = ConnectionState.Open Then conn.Close()
disrecords()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
DateTimePicker2.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
TextBox1.Focus()
End Sub

```

```

Sub disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from CarRentals", conn)
adp.Fill(DS1)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

```

```

Private Sub Butsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
saveRecord()
Butnew.Enabled = True
Butsave.Enabled = True
End Sub

```

```

Private Sub Short_Trip_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
disrecords()

```

```

TextBox12.Text = Format(Today(), "dd/MM/yyyy")
Me.Size = TTDBMDI.Size
End Sub

```

```

Private Sub Butsave_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Butsave.Click
saveRecord()
Butnew.Enabled = True
Butsave.Enabled = True
End Sub

```

```

Private Sub Butnew_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Butnew.Click
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
DateTimePicker2.Text = ""
ComboBox1.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
TextBox1.Focus()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("Select max(PassengerID) from CarRentals", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
TextBox1.Text = IIf(IsDBNull(d1(0)), 100, d1(0)) + 1
Else
TextBox1.Text = "101"
End If
TextBox1.Enabled = False
Butnew.Enabled = False
Butsave.Enabled = True
End Sub

```

```

Private Sub Butexit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Butexit.Click
Me.Close()
End Sub

```

```

Private Sub TextBox9_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TextBox9.TextChanged
If Val(TextBox9.Text) <= 100 Then
TextBox10.Text = 200

```

```

ElseIf Val(TextBox9.Text) > 100 And Val(TextBox9.Text) <= 300 Then
    TextBox10.Text = 300
ElseIf Val(TextBox9.Text) > 300 And Val(TextBox9.Text) <= 500 Then
    TextBox10.Text = 500
ElseIf Val(TextBox9.Text) > 500 And Val(TextBox9.Text) <= 1000 Then
    TextBox10.Text = 1000
ElseIf Val(TextBox9.Text) > 1000 Then
    TextBox10.Text = 2000
End If
If ComboBox1.Text = "INDICA VISTA" Then
    TextBox11.Text = 8.5 * Val(TextBox9.Text)
ElseIf ComboBox1.Text = "SWIFT DZIRE" Then
    TextBox11.Text = 9 * Val(TextBox9.Text)
ElseIf ComboBox1.Text = "ETIOS" Then
    TextBox11.Text = 11 * Val(TextBox9.Text)
ElseIf ComboBox1.Text = "INNOVA CRYSTA" Then
    TextBox11.Text = 14 * Val(TextBox9.Text)
ElseIf ComboBox1.Text = "ERTIGA" Then
    TextBox11.Text = 11 * Val(TextBox9.Text)
    TextBox11.Show()
End If
End Sub

```

```

Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton1.CheckedChanged
    TextBox3.Text = "MALE"
End Sub

```

```

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton2.CheckedChanged
    TextBox3.Text = "FEMALE"
End Sub

```

```

Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton3.CheckedChanged
    TextBox3.Text = "OTHERS"
End Sub

```

```

Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged
    TextBox3.Enabled = False
End Sub
End Class

```

DEVOTIONAL RECORDS CODING:

```

Imports System.Data.SqlClient
Public Class DEVOTIONAL_TOUR
Sub disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from Bookings where Package='DEVOTIONAL
PACKAGES' ", conn)
adp.Fill(DS1)
DG1.DataSource = DS1.Tables(0)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Me.Close()
End Sub

Private Sub DEVOTIONAL_TOUR_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Me.Size = TTDBMDI.Size
TextBox12.Text = Format(Today(), "dd/MM/yyyy")
disrecords()
End Sub
End Class

```

CANCELLATION CODING:

```

Imports System.Data.SqlClient
Imports System.Text.RegularExpressions

Public Class Devotional_Cancellation
Dim pkvar As String
Sub saveRecord()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
q1var = "Insert into Cancellation("
q2var = "values("
q1var = q1var & "PassengerID" & ","
q2var = q2var & "" & ComboBox1.Text & ","
q1var = q1var & "Name" & ","
q2var = q2var & "" & UCASE(TextBox2.Text) & ","
q1var = q1var & "Gender" & ","
q2var = q2var & "" & TextBox3.Text & ","
q1var = q1var & "Contact" & ","
q2var = q2var & "" & Val(TextBox4.Text) & ","
q1var = q1var & "EmailID" & ","
q2var = q2var & "" & TextBox5.Text & ","
q1var = q1var & "Add1" & ","
q2var = q2var & "" & TextBox6.Text & ","
q1var = q1var & "Add2" & ","

```

```

q2var = q2var & "" & TextBox7.Text & ","
q1var = q1var & "Add3" & ","
q2var = q2var & "" & TextBox8.Text & ","
q1var = q1var & "Date" & ","
q2var = q2var & "" & TextBox9.Text & ","
q1var = q1var & "Days" & ","
q2var = q2var & "" & ComboBox3.Text & ","
q1var = q1var & "Total" & ","
q2var = q2var & "" & TextBox10.Text & ","
q1var = q1var & "DOR" & ","
q2var = q2var & "" & DateTimePicker1.Text & ","
q1var = q1var & "Refund" & ")"
q2var = q2var & "" & TextBox11.Text & ")"
Dim cmd1 As New SqlCommand(q1var & q2var, conn)
cmd1.ExecuteNonQuery()
If conn.State = ConnectionState.Open Then conn.Close()
disrecords()
ComboBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
ComboBox3.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
ComboBox1.Focus()
End Sub
Sub disrecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select
PassengerID,Name,Gender,Contact,EmailID,Add1,Add2,Add3,Date,Days,Total
from
Bookings", conn)
adp.Fill(DS1)
If conn.State = ConnectionState.Open Then conn.Close()
TextBox2.Enabled = False
TextBox3.Enabled = False
TextBox4.Enabled = False
TextBox5.Enabled = False
TextBox6.Enabled = False
TextBox7.Enabled = False
TextBox8.Enabled = False
TextBox9.Enabled = False
ComboBox3.Enabled = False
TextBox10.Enabled = False
TextBox11.Enabled = False

```

```

End Sub
Private Sub Butexit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
End
End Sub

Private Sub SouthIndia_Cancellation_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Me.Size = TTDBMDI.Size
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select PassengerID from Bookings order by PassengerID",
conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader
While d1.Read
ComboBox1.Items.Add(d1(0).ToString)
End While
disrecords()
Static count As Integer
count = count + 1
Select Case count
Case 1
PictureBox5.BringToFront()
Case 2
PictureBox4.BringToFront()
Case 3
PictureBox2.BringToFront()
End Select
If count = 3 Then
count = 0
End If
End Sub

Private Sub TextBox11_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
TextBox11.Text = Val(TextBox10.Text) * 0.75
End Sub

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Me.Close()
End Sub

Private Sub Butcancel_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Butcancel.Click
saveRecord()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
q1var = "Insert into Cancellation("
q2var = "values("
q1var = q1var & "PassengerID" & ","
q2var = q2var & "" & ComboBox1.Text & ","
q1var = q1var & "Name" & ","

```

```

q2var = q2var & "" & UCASE(TextBox2.Text) & ","
q1var = q1var & "Gender" & ","
q2var = q2var & "" & TextBox3.Text & ","
q1var = q1var & "Contact" & ","
q2var = q2var & "" & Val(TextBox4.Text) & ","
q1var = q1var & "EmailID" & ","
q2var = q2var & "" & TextBox5.Text & ","
q1var = q1var & "Add1" & ","
q2var = q2var & "" & TextBox6.Text & ","
q1var = q1var & "Add2" & ","
q2var = q2var & "" & TextBox7.Text & ","
q1var = q1var & "Add3" & ","
q2var = q2var & "" & TextBox8.Text & ","
q1var = q1var & "Date" & ","
q2var = q2var & "" & TextBox9.Text & ","
q1var = q1var & "Days" & ","
q2var = q2var & "" & ComboBox3.Text & ","
q1var = q1var & "Total" & ","
q2var = q2var & "" & TextBox10.Text & ","
q1var = q1var & "DOR" & ","
q2var = q2var & "" & DateTimePicker1.Text & ","
q1var = q1var & "Refund" & ")"
q2var = q2var & "" & TextBox11.Text & ")"
If vbNo = MsgBox("ARE YOU SURE YOU WANT TO CANCEL YOUR TICKET",
MsgBoxStyle.YesNo, " Delete") Then Exit Sub
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd1 As New SqlCommand("Delete from Bookings where PassengerID =" &
Val(ComboBox1.Text) & "", conn)
cmd1.ExecuteNonQuery()
If conn.State = ConnectionState.Open Then conn.Close()
disrecords()
ComboBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
ComboBox3.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
ComboBox1.Focus()
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()

```

```

Dim cmd0 As New SqlCommand("select
PassengerID,Name,Gender,Contact,EmailID,Add1,Add2,Add3,Date,Days,Total
from
Bookings where PassengerID=''" & ComboBox1.Text & "", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
ComboBox1.Text = d1(0).ToString
TextBox2.Text = d1(1).ToString
TextBox3.Text = d1(2).ToString
TextBox4.Text = d1(3).ToString
TextBox5.Text = d1(4).ToString
TextBox6.Text = d1(5).ToString
TextBox7.Text = d1(6).ToString
TextBox8.Text = d1(7).ToString
TextBox9.Text = d1(8).ToString
ComboBox3.Text = d1(9).ToString
TextBox10.Text = d1(10).ToString
TextBox11.Text = Val(TextBox10.Text) * 0.75
TextBox11.Show()
End If
End Sub

```

```

Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox3.Click
Static count As Integer
count = count + 1
Select Case count
Case 1
PictureBox5.BringToFront()
Case 2
PictureBox4.BringToFront()
Case 3
PictureBox2.BringToFront()
End Select
End Sub

```

```

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged
If (TextBox1.Text = "XDHYN") Then
Butcancel.Enabled = True
Else
Butcancel.Enabled = False

If (TextBox1.Text = "WDFUC") Then
Butcancel.Enabled = True
Else
Butcancel.Enabled = False

If (TextBox1.Text = "e5hb") Then
Butcancel.Enabled = True
Else

```

```

Butcancel.Enabled = False
End If
End If
End If
End Sub
End Class

```

GOA REPORT CODING;

```

Imports System.Data.SqlClient
Public Class Goa_Report
Sub display()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim ds1 As New DataSet
Dim adp As New SqlDataAdapter("Select PassengerID from Bookings order by PassengerID ", conn)
adp.Fill(ds1)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
Dim xpos As Integer, ypos As Integer
ypos = 50
Dim MyFont As New Font("Impact", 20)
xpos = 250
e.Graphics.DrawString("GOA REPORT DETAILS", MyFont, Brushes.Blue, xpos, ypos)
ypos = 150
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("Select PassengerID,Name,Gender,Contact,EmailID,Add1,Add2,Add3,Date,Days,TOTAL,Package from Bookings where PassengerID=''' & ComboBox1.Text & '''", conn)
Dim d2 As SqlDataReader = cmd0.ExecuteReader()
While d2.Read
xpos = 130
MyFont = New Font("Arial", 18)
e.Graphics.DrawString("Passenger ID", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(0).ToString, MyFont, Brushes.Black, xpos, ypos)
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Name", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(1).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Address", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300

```

```
e.Graphics.DrawString(d2(2).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(3).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(4).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(5).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Contact", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(6).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Email ID", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(7).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Date", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(8).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Days", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(9).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
e.Graphics.DrawString("Total", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(10).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
```

```

xpos = 130
e.Graphics.DrawString("Places", MyFont, Brushes.Black, xpos, ypos)
xpos = xpos + 300
e.Graphics.DrawString(d2(11).ToString, MyFont, Brushes.Black, xpos, ypos)
xpos = 50
ypos = ypos + 80
xpos = 130
End While
End Sub

```

```

Private Sub PP1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
PP1.Show()
End Sub

```

```

Private Sub Goa_Report_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("Select PassengerID from Bookings order by PassengerID ", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader
While d1.Read
ComboBox1.Items.Add(d1(0).ToString)
End While
display()
End Sub

```

```

Private Sub ButShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButShow.Click
PP1.Show()
End Sub

```

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("Select * from Bookings where PassengerID ='" &
ComboBox1.Text & "'", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
ComboBox1.Text = d1(0).ToString
End If
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
End
End Sub
End Class

```

PRINT FORM CODING:

```

Imports System.Data.SqlClient
Imports System.Text.RegularExpressions
Imports System.Text
Imports System.Drawing
Imports System.Windows.Forms
Imports System.Drawing.Drawing2D
Public Class PRINT
Dim pkvar As String

Sub disRecords()
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim DS1 As New DataSet
Dim adp As New SqlDataAdapter("select * from Bookings where PassengerID=" & pkvar &
" ", conn)
adp.Fill(DS1)
If conn.State = ConnectionState.Open Then conn.Close()
End Sub

Private Sub PRINT_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select PassengerID from Bookings order by PassengerID", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader
While d1.Read
ComboBox1.Items.Add(d1(0).ToString)
End While
disRecords()
End Sub

Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
Dim xpos As Integer, ypos As Integer
PP1.MdiParent = TTDBMDI
ypos = 50
xpos = 10
Dim MyFont As New Font("GABRIOLA BOLD", 26)
xpos = 160
e.Graphics.DrawString("INCREDIBLE INDIA TRAVELS", MyFont, Brushes.Red, xpos, ypos)
ypos += 60
MyFont = New Font("Times New Roman", 20)
xpos = 230
e.Graphics.DrawString("One country many worlds...!", MyFont, Brushes.Black, xpos, ypos)
ypos = ypos + 30
xpos = 0
e.Graphics.DrawString(" _____", MyFont, Brushes.BlueViolet, xpos, ypos)

```

```

ypos += 5
xpos = 1
MyFont = New Font("impact", 30)
xpos = 80
ypos = 210
e.Graphics.DrawString("PASSENGER DETAILS", MyFont, Brushes.Cyan, xpos, ypos)
ypos += 100
ypos += 2
xpos = 10
MyFont = New Font("Georgia Bold", 20)
xpos = 100
ypos = 307
e.Graphics.DrawString("NAME", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = 100
MyFont = New Font("Times New Roman ", 18)
e.Graphics.DrawString(textBox2.Text, MyFont, Brushes.Black, xpos, ypos)
xpos = 10
ypos = 395
MyFont = New Font("Georgia Bold", 20)
xpos = 90
ypos += 38
e.Graphics.DrawString("CONTACT NUMBER", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = 90
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString(textBox4.Text, MyFont, Brushes.Black, xpos, ypos)
ypos += 1
ypos = 305
xpos = 400
MyFont = New Font("Georgia Bold", 20)
xpos = xpos + 100
ypos += 8
e.Graphics.DrawString("GENDER", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = xpos + 4
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString(textBox3.Text, MyFont, Brushes.Black, xpos, ypos)
ypos = ypos + 70
ypos += 1
ypos = 382
xpos = 393
MyFont = New Font("Georgia Bold", 20)
xpos = xpos + 105
ypos += 50
e.Graphics.DrawString("EMAIL ID", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = xpos + 7
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString(textBox5.Text, MyFont, Brushes.Black, xpos, ypos)
ypos = ypos + 70

```

```

xpos = 0
ypos += 1
e.Graphics.DrawString("_____", MyFont, Brushes.Blue, xpos, ypos)
MyFont = New Font("impact", 30)
xpos = 80
ypos = 620
e.Graphics.DrawString("PACKAGE DETAILS", MyFont, Brushes.Cyan, xpos, ypos)
xpos = 10
ypos = 680
MyFont = New Font("Georgia Bold", 20)
xpos = 90
ypos += 38
e.Graphics.DrawString("BOOKING STATUS", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = 90
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString("Confirmed", MyFont, Brushes.Black, xpos, ypos)
ypos = 710
xpos = 400
MyFont = New Font("Georgia Bold", 20)
xpos = xpos + 100
ypos += 8
e.Graphics.DrawString("DATE OF TRIP", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = xpos + 4
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString(DateTimePicker2.Text, MyFont, Brushes.Black, xpos, ypos)
ypos = ypos + 70
xpos = 10
ypos = 800
MyFont = New Font("Georgia Bold", 20)
xpos = 90
ypos += 38
e.Graphics.DrawString("NO. OF DAYS", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = 90
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString(ComboBox1.Text, MyFont, Brushes.Black, xpos, ypos)
ypos = 830
xpos = 400
MyFont = New Font("Georgia Bold", 20)
xpos = xpos + 100
ypos += 8
e.Graphics.DrawString("TOTAL PRICE", MyFont, Brushes.Blue, xpos, ypos)
ypos = ypos + 50
xpos = xpos + 4
MyFont = New Font("Times New Roman", 18)
e.Graphics.DrawString(TextBox9.Text, MyFont, Brushes.Black, xpos, ypos)
ypos = ypos + 70
xpos = 10
ypos = 900

```

```

MyFont = New Font("Georgia Bold", 20)
xpos = 0
ypos += 38
e.Graphics.DrawString(" _____ ", MyFont, Brushes.MediumBlue, xpos, ypos)
ypos = ypos + 70
xpos = 125
MyFont = New Font("georgia Bold", 20)
e.Graphics.DrawString("We are not responsible for your belongings", MyFont, Brushes.Black,
xpos, ypos)
ypos = ypos + 80
xpos = 220
MyFont = New Font("gabriola ", 26)
e.Graphics.DrawString("HAPPY JOURNEY...", MyFont, Brushes.Red, xpos, ypos)
End Sub

```

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
If conn.State = ConnectionState.Open Then conn.Close()
conn.Open()
Dim cmd0 As New SqlCommand("select
PassengerID,Name,Gender,Contact,EmailID,Add1,Add2,Add3,Date,Days,Total
from
Bookings where PassengerID=''' & ComboBox1.Text & '''", conn)
Dim d1 As SqlDataReader = cmd0.ExecuteReader()
If d1.HasRows Then
d1.Read()
ComboBox1.Text = d1(0).ToString
TextBox2.Text = d1(1).ToString
TextBox3.Text = d1(2).ToString
TextBox4.Text = d1(3).ToString
TextBox5.Text = d1(4).ToString
TextBox6.Text = d1(5).ToString
TextBox7.Text = d1(6).ToString
TextBox8.Text = d1(7).ToString
DateTimePicker2.Text = d1(8).ToString
ComboBox2.Text = d1(9).ToString
TextBox9.Text = d1(10).ToString
PP1.Show()
End If
End Sub

```

```

Private Sub PP1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles PP1.Load
PP1.Show()
End Sub
End Class

```

CHAPTER: 10

TESTING AND RESULTS

10.1 SOFTWARE TESTING

As the coding is completed according to the requirement, we have to test the quality of the software. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Although testing is to uncover the errors in the software but it also demonstrates that software functions appear to be working as per the specification, those performance requirements appear to have been met. In addition, data collected as testing is conducted provide a good indication of software reliability and some indications of software quality as a whole.

To assure the software quality we conduct both White Box Testing and Black Box Testing.

10.2 WHITE BOX TESTING:

White Box Testing is a test case design method that uses the control structure of the procedural designs to derive test cases. As we are using a non procedural language, there is very small scope for the white Box Testing. Whenever it is necessary, there the control structures are tested and successfully passed all the control structures with a very minimum error.

10.3 BLACK BOX TESTING:

Black Box Testing focuses on the functional requirement of the software. It enables to derive sets of input conditions that will fully exercise all functional requirements for a program.

The Black Box Testing finds almost all errors. If finds some interface errors and errors in accessing the database and some performance errors. In Black Box Testing we use mainly two techniques: Equivalence Partitioning the Boundary Volume Analysis Technique.

10.4 Equivalence Partitioning:

In this method we divide input domain of program into classes of data from which test cases are derived. An Equivalence class represents a set of valid or invalid or a set of related values or a Boolean condition. The equivalence for these is:

Input condition requires specific, value-specific or non-specific two classes.

- Input condition requires a range- in the range or out of range two classes.
- Input condition specifies a member of a set- belongs to a set or not belongs to the set two classes.
- Input condition is Boolean- valid or invalid Boolean condition two classes.

By these types of equivalent classes, we can test for many cases.

10.5 Boundary Values Analysis:

Number of errors usually occurs at the boundaries of the input domain generally. In this technique a selection of test cases is exercised using boundary values i.e., around boundaries. By the above two techniques, we eliminated almost all errors from the software and checked for numerous test values for each and every input value. The results were satisfactory.

10.6 System Code:

System testing is designated to uncover weakness that was not detected in the earlier tests. The total system is tested for recovery and fallback after various major failures to ensure that no data are lost. An acceptance test is done to validity and reliability of the system. The philosophy behind the testing is to find error in project. There are many test cases designed with this in mind. The flow of testing is as follows:

10.6.1 Code Testing:

Specification testing is done to check if the program does what it should do and how it should behave under various condition or combinations and submitted for processing in the system and it is checked if any overlaps occur during the processing.

This strategy examines the logic of the program. Here only syntax of the code is tested. In code testing syntax error are corrected, to ensure that the code is perfect.

10.6.2 Unit Testing:

The first level of testing is called unit testing. Here different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test oracles.

Unit testing comprises a set of tests preformed by an individual programmer prior to integration of the units into a large system. A program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on the modules to locate errors. These errors are verified and corrected so that the unit perfectly fits to the project.

10.6.3 System Testing:

The next level of testing is system testing and acceptance testing. This testing is done to check if the system has met its requirements and to find the external behavior of the system.

System testing involves two kinds of activities:

- Integration testing
- Acceptance testing.

10.6.4 Integration Testing:

The next level of testing is called the Integration Testing. In this many tested modules are combined into subsystems, which were then tested.

Test case data is prepared to check the control flow of all the modules and to exhaust all possible inputs to the program. Situations like treating the modules when there is no data entered in the test box is also tested.

This testing strategy dictates the order in which modules must be available, and exerts strong influence on the order in which the modules must be written. Debugged and unit tested. In

integration testing, all the modules/units on which unit testing is performed are integrated together and tested.

10.6.5 Acceptance Testing:

This testing is performed finally by user to demonstrate that the implemented system satisfies its requirements. The user gives various inputs to get required outputs.

10.6.6 Specification Testing:

specification testing is done to check if the program does what it should do and how it should behave under various conditions or combination and submitted for processing in the system and it is checked if any overlaps occur during the processing.

10.6.7 Performance Time Testing:

Performance time testing is done to determine how long it takes to accept and respond i.e., the total time for processing when it has to handle quite a large number of records. It is essential to check the exception speed of the system, which runs well with only a handful of test transactions. Such systems might be slow when fully loaded. So testing is done by providing large amount of data for processing. A system testing is designed to uncover weaknesses that were not detected in the earlier tests.

The total system is tested for recovery and fallback after various major failures to ensure that no data are lost during an emergency. An acceptance test is done to ensure the user about the validity and reliability of the system.

10.7 Tours and Travels:

Following are some of the important test cases and expected outcomes as per the testing plan used for the Tours and Travels.

1. Tests conducted in the activities of the Administrator Module are as follows:

Test Case No.	1
Module	Administrator-General
User Form	Login Form-enter the login-id and user name
Input	Click submit button without entering login-id
Expected Result	Display a message saying, "Please enter the login-id."
Test Results	Tested – OK

Test Case No.	2
Module	Administrator-To See Booking form
User Form	Entry form.
Input	Click submit button to book ticket.
Expected Result	Display a message saying, "YOUR TICKET HAS BEEN BOOKED SUCCESSFULLY"
Test Results	Tested – OK

2. Tests conducted in the activities of the Ticket Cancellation are as follows:

Test Case No.	3
Module	Ticket cancellation
User Form	Main Menu
Input	Click cancel button to cancel ticket.
Expected Result	Display a message saying, "Are you sure about cancelling your ticket"
Test Results	Tested – OK

CHAPTER NO: 11**CONCLUSIONS**

The system is designed in such an extendable fashion to incorporate the future changes into the system easily. Any particular can be added easily and get connected to the menu. The various user-friendly features are introduced in this project. There are several future ideas to develop this application, still in mind. All the links to other forms are generated and are tested with the actual system.

The database connectivity is tested with the user requirements and verified for the validity. The software requirements have been met. Needed documents are generated and adequate documentation has been provided for maintenance and further enhancements. The operation required by the user to operate the system is basic computer knowledge only.

Finally, this is Well graphics project and all the forms are very easy to understand for every computer operator, so that the transactions of the databases are completely recorded in a report in the form of SQL data report. No confusions for the required Final printing reports for all modules in well apped project such as tour details, car rentals details, north india package booking, south india package booking, goa package booking, devotional package booking, car rentals booking, modify package ticket, search tickets, north india show records, south india show records, goa show records, devotional show records, ticket cancellation, cancellation records, print ticket, north india report, south india report, goa report, devotional report and about us. These modules are completely validated and 100% verifications done in the frontend tool visual studio and inserted well graphics design.

CHAPTER NO: 12**FUTURE ENHANCEMENTS**

Enhancement means adding, modifying, or redeveloping the code to support changes in the specifications. It is necessary to keep up with changing user needs and the operational environment.

Since no system is ever complete, this app is useful for reserving the tickets for passengers further developed to include more operations and analysis, as changes are required by end users of the system.

The System can also be developed using VB.NET with the help of Application software.

The System can also be developed for all Police station maintenance.

Graphical Reports can be designed with graphs such as data flow diagram and ER diagram. This will provide clear picture of a ratio of the stored data can be retained for future enhancement.

CHAPTER NO: 13**BIBLIOGRAPHY****REFERENCE BOOKS:**

Head first Visual Basic – by Basham, Sierra and Bates.

Complete Reference – 5th Edition fifth Edition by Herbert Schildt, Published by McGraw-Hill Publishing Company Limited.

SoftwareEngineering - A Practitioner's Approach, Fifth Edition by Roger S Pressman, Published by McGraw-Hill International Edition.

Beginning Visual Basic 2010– By Kelvin Muker and James L weaver with James P crume published by a! Express

Data base – By Elmasri Navathe