SDJ INTERNATIONAL COLLEGE

Bachelor of Computer Applications (BCA) Programme

Project Report

BCA Sem IV

AY 2021-22

Airline Reservation System

by

Roll No.	Name of Student
095	KAPADIA MANSI SHAILESHKUMAR

Project Guide by :

Prof.Nidhi Desai

SDJ INTERNATIONAL COLLEGE

CERTIFICATE

This is to certify that Ms. KAPADIA MANSI SHAILESHKUMAR examination number 095 has satisfactorily completed her project work entitled Airline Reservation System as partial fulfillment of requirements for BCA Sem IV, during the academic year 2021-22.

Date:

Place: Surat

Prof.Nidhi Desai
SDJ International College,
Surat

INDEX

Introduction

1.1 Project Summary

This is an Airline Reservation System project for the learner of Vb.net. The system has 2 sides of users which are the Admin and the User. The Admin user is in charge of managing the Flight, Booking, Scheduling, and Reports. The Regular User will be the one who can **Login** and **Reserve** a flight for the passengers and also the one who can cancel the reservation of the passengers. This Project using MS - ACCESS for the database. The system is user-friendly and has a simple user interface.

Independent travellers generally have only a rough idea of where they want to travel, how long they intend to spend in each destination and which attractions they intend to visit before they depart. A dynamic **Airline Reservation System** that is able to dynamically modify the traveler's information in response to events that occur during the trip such as the travellers changing location, availability of new last-minute specials, flight details and safety issues could provide booked ticket flexibility. This describes architecture for such a system based on distributed co operating software agents and technologies.

1.2 Project Technical Profile

Software Requirements

Software Configuration	
Operating System	Window 10 or Windows 11
Front – End	VB.NET
Back – End	MS - ACCESS

Hardware Requirement

Hardware Configuration	
Processor	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
RAM	8.00 GB
Hard Disk	512 SSD
System Type	64 – bit Operating System, x64 – based Processor

Scope & Planning

SCOPE

- Airline Reservation System is Booking Desktop Application.
- This Desktop Application is developing with the purpose of providing Login, Flight Information and booking.
- Unregister user can't enter in System.
- It provides different flights information in almost different country.
- Get various Flight details fare information.
- Get various source and destination different time of flight.
- View latest and best deal on the day.
- The site will help to estimate the benefits and calculate the requirement very easily.
- Registered User can book flights, View Flights, Booking History.

Objective

- The creation of a unique, upscale, innovative and tour Packages that will differentiate from other sites.
- To serve its customers with best possible quality and provide them maximum
facilities with less price.
- To provide transportation with best possible quality and comfort as per our

customers need.

- Ensure Customer satisfaction.

2.1 Requirement Analysis

Success of any system depends mostly on how well Data/Requirement gathering are done. It provides direction to system analyst and designers to design a system that is efficient.

- To gather requirement for Airline Reservation System, following steps were carried out We got several information from Prof. **Mrs. Nidhi Desai** (as our Project Guide). She helped us to understand required functionalities, also noted down some useful features to be included and also to know what about input and output.
- We also consulted senior experts for technical problems.
- We also visited websites for getting knowledge about existing system.
- E.g.https://en.wikipedia.org/wiki/Airline_reservations_system

Understanding the Process of System:

• How to Manage all Forms?

• How to Manage Login and Registration Process?

• How to know How Many users are Created and How many Queries are Generated on the Particular Day?

• How to Manage All Forms linked with Each Other?

• How to Manage the Bunch of a Particular Member So that It can have the Security purpose and Keep the Data Private?

• How to Perform All User Security Level and Perform A Authentication and Give Personal Account to each User for Security Purpose?

➢USER-SITE:

Our model is Diligently Focused on Ensuring a Positive Customer Experience That in Booking Flight across over world.

✓User can Registered and Login in System.

✓System should provide facility to Inquiry from admin.

✓ System should provide advanced flight searching.

✓ System should provide facility manage Booked flights.

✓ System should provide facility to manage Booking and History.

✓ System should provide booking of Flight.

➢ADMIN-SITE:

System should provide following information to Admin.

• Registered User

• Registered New Admin

• Admin Login

• Add Flight Details

• Manage User Booking

2.2 Feasibility Study

A feasibility study is determined the possibility or probability of either improving the existing system or developing a completely new system. It helps to obtain an overview of the problem and to get rough assessment of whether feasible solution exists.

There are many aspects in feasibility study portion of the preliminary investigation.

Technological Feasibility:-

• The technical Feasibility work for the project is done with the present equipment, manual procedures, and existing software technology & available technology hardware.

• The necessary technology, front-end, backend database technology, and other tools namely reporting tool, installation tools, etc., (as mentioned in the project profile) for developing the system, are already available within the organization.

Economical Feasibility:-

• Economic feasibility looks at the financial aspects of the project.

• Economic feasibility concerns with the returns from the investment in a project.

• It determines whether it is worthwhile to invest the money in the proposed system. It is not worthwhile spending a lot of money on a project for no returns.

• To carry out an economic feasibility for a system, it is necessary to place actual money value against any purchases or activities needed to implement the project.

Operational Feasibility:-

- This phase is system operations performed by proposed system in this phase here.

- Testing is the important point is user will be operating the system.

- System will work under any condition .System will allow entering & retrieving of Data only to authorized user to operate the system.

- It will work any full efficiency & accuracy as used to work with particular computer.

- There is sufficient support to the project from the management and from the intended users of the system.

2.3 Future Development

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- Provide More Security.

- Add More Functionality in Searching for Reserve Ticket.

- Good facility for forgot password.

- User can check the privacy setting for their account.

- User can check the change password.

- New Option Display Reports.

Designing

3.1 DFD (Data Flow Diagram)

Level 0 System

Admin 1st Level

User 1st Level

3.2 Use Case Diagram

User Functionality

Admin Functionality

3.3 Entity - Relationship Diagram

3.4 Database Design

Table Name :admin_master

Description :This table is give details about admin information

Field Name	Field Type	Constraint	Description
Admin_id	Autonumber	PrimaryKey	Id
Admin_Name	Text	Not Null	Name of Admin
Admin_DOB	Date/Time	Not Null	Date of Birth
Admin_Adharcard	Number	Not Null	Adharcard Number
Admin_contact	Number	Not Null	Contact Info
Admin_emailid	Text	Not Null	Email id

Admin_password	Number	Not Null	Password

Table Name :book

Description : This table is for booking flight for user.

Field Name	Field Type	Constraint	Description
Book_id	Autonumber	PrimaryKey	Id of Book
User_id	Number	Foreign Key	User id
Flight_id	Number	Foreign Key	Id of Flight
Flight_Date	Date/Time	Not Null	Flight Date
Flight_To	Text	Not Null	Source
Flight_From	Text	Not Null	Destination
Ticket_Fare	Number	Not Null	Price
Ticket_Seat	Number	Not Null	No of Seat
Total_Fare	Number	Not Null	Total Cost

Table Name :flight_master

Description : This table is give details about flights information

Field Name	Field Type	Constraint	Description
Flight_id	Autonumber	PrimaryKey	Id
Flight_Name	Text	Not Null	Flight Name
Flight_From	Text	Not Null	Source
Flight_To	Text	Not Null	Destination
Flight_Date	Text	Not Null	Date of Flight
Flight_Hour	Number	Not Null	Hour
Flight_Amount	Number	Not Null	Amount

Table Name :user_master

Description : This table is for Details of User.

Field Name	Field Type	Constraint	Description
User_Id	Autonumber	PrimaryKey	Id of user
User_FirstName	Text	Not Null	First Name
User_Name	Text	Not Null	Middle Name
User_LastName	Text	Not Null	Last Name
User_Address	Memo	Not Null	Address
User_Gender	Text	Not Null	Gender
User_DOB	Date/Time	Not Null	Date of Birth
User_Age	Number	Not Null	Age of User
User_Emailid	Text	Not Null	Email id

User_contact	Number	Not Null	Contact Number
User_Pancard	Number	Not Null	Pan Card Number
User_Passport	Number	Not Null	Passport Number
User_Adharcard	Number	Not Null	Adhar card Number
User_password	Number	Not Null	Password

3.5 User Interface

Login Form

Description: Main form from user and admin can login

Coding

```
PublicClasslogin

PrivateSub Button1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button1.Click
'showadmin_log
Dim x AsNewadmin_log
```

```
x.Show()
Me.Hide()
EndSub

PrivateSub Button2_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button2.Click
'showuser_login
Dim x AsNewuser_login
x.Show()
Me.Hide()
EndSub


EndClass
```

Admin Login

Description: Admin Login Form

Coding

```
ImportsSystem.Data.OleDb
PublicClassadmin_log
'admin_login objects
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
DimdrAsOleDbDataReader

'admin_master login
PrivateSubbtncreaccount_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handlesbtncreaccount.Click
Dim x AsNewadmin_master
x.Show()
Me.Hide()

EndSub

PrivateSubbtn_login_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtn_login.Click
'insert of admin_name and password
cn.Open()
qry = "select * from admin_master where Admin_Name='"&txtadmin_name.Text&"' and
Admin_password="&txtadmin_pass.Text

cmd = NewOleDbCommand(qry, cn)
dr = cmd.ExecuteReader
Ifdr.HasRowsThen
dr.Read()
```

```
Dim x AsNewadmin_MDI
x.Show()
Me.Hide()
Else
MsgBox("U HAVE NOT SIGN UP THEN U CAN LOGIN..... ", MsgBoxStyle.Critical, "NOT
FOUND ....")
EndIf
cn.Close()
EndSub

EndClass
```

Admin Master Form (SignUp)

Description: This form is to create new Admin

Coding

```
ImportsSystem.Data.OleDb

PublicClassadmin_master
'admin_master object
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
Dim dap AsNewOleDbDataAdapter
Dim ds AsDataSet

'insert ofadminadmin_id,adminname,DOB,Adharcard,contact,email_id,password
PrivateSubbtnsubmit_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtnsubmit.Click
cn.Open()
qry = "insert into
admin_mastervalues("&txtadmin_id.Text&",'"&txtadmin_name.Text&"','"&datetimepickerdob.
Text&"',"&txtadadharcard.Text&","&txtadcontact.Text&",'"&txtademail.Text&"',"&txtadmin_pa
ssword.Text&")"
cmd = NewOleDbCommand(qry, cn)
cmd.ExecuteNonQuery()
cn.Close()
MsgBox("RECORD INSERTED SUCCESSFULLY.......", MsgBoxStyle.Information)
Dim x AsNewadmin_log
x.Show()
Me.Close()
EndSub
EndClass
```

Admin Home Page

Description: Admin Home Page after login

Coding

```
ImportsSystem.Windows.Forms

PublicClassadmin_MDI

PrivateSubViewBookingToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesViewBookingToolStripMenuItem.Click
'object of viewbook and show
Dim x AsNewviewbook
x.Show()

EndSub

PrivateSubViewUserToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesViewUserToolStripMenuItem.Click
'object of viewuser  and show
Dim x AsNewviewuser
x.Show()

EndSub

PrivateSubFlightDetailsToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesFlightDetailsToolStripMenuItem.Click
'object of  filght_master and show
Dim x AsNewfilght_master
x.Show()
EndSub

PrivateSubAdminMasterToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)
'object of admin_master  and show
Dim x AsNewadmin_master
x.Show()

EndSub

PrivateSubAdminLoginToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)
'object of admin_log and show
Dim x AsNewadmin_log
x.Show()
EndSub


EndClass
```

Flight Master

Description: Admin Home Page after login

Coding

```
ImportsSystem.Data.OleDb
PublicClassfilght_master
'object of flight_master
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
DimadpAsNewOleDbDataAdapter
Dim ds AsNewDataSet

PrivateSubfilght_master_Load(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesMyBase.Load
fillgrid()
clear()

EndSub

PrivateSubbtn_insert_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtn_insert.Click
'insert of flight id,name,form,to,date,hour,amount
cn.Open()
qry = "insert into
flight_mastervalues("&txtid.Text&",'"&txtname.Text&"','"&ComboBox_from.Text&"','"&Combo
Box_to.Text&"','"&DateTimePickerdate.Text&"',"&txtf_hour.Text&","&txtf_amt.Text&")"
cmd = NewOleDbCommand(qry, cn)
cmd.ExecuteNonQuery()
cn.Close()
MsgBox("RECORD INSETED SUCCCESSFULLY...")
fillgrid()
clear()

EndSub
Subfillgrid()
'data adapter
ds.Clear()
qry = "select * from flight_master"
adp = NewOleDbDataAdapter(qry, cn)
```

```
adp.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)

EndSub
Subclear()
'clear
Dim x AsControl
ForEach x InMe.Controls
IfTypeOf x IsTextBoxThen
x.Text = ""
EndIf
Next

EndSub
DimdrAsOleDbDataReader
Dim fid AsInteger

PrivateSubbtnsearch_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtnsearch.Click
'search of flight id,name,form,to,date,hour,amount
        fid = InputBox("Enter Flight ID : ")
cn.Open()
qry = "select * from flight_master where Flight_Id="& fid
cmd = NewOleDbCommand(qry, cn)
dr = cmd.ExecuteReader()
Ifdr.HasRowsThen
dr.Read()
txtid.Text = dr("Flight_Id")
txtname.Text = dr("Flight_Name")
ComboBox_from.Text = dr("Flight_From")
ComboBox_to.Text = dr("Flight_To")
DateTimePickerdate.Text = dr("Flight_Date")
txtf_hour.Text = dr("Flight_Hour")
txtf_amt.Text = dr("Flight_Amount")
EndIf
cn.Close()
EndSub
DimansAsString

PrivateSubbtndel_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtndel.Click
'delete of flight id,name,form,to,date,hour,amount
ans = MsgBox("Are U sure U Want To Delete", MsgBoxStyle.Exclamation +
MsgBoxStyle.YesNo, "My Delete Message...")
Ifans = vbYesThen

cn.Open()
qry = "delete from flight_master where Flight_Id="&txtid.Text
cmd = NewOleDbCommand(qry, cn)
cmd.ExecuteNonQuery()
cn.Close()
fillgrid()
clear()

Else
```

```
MsgBox("Thank U...")
EndIf
EndSub

PrivateSubbtnupdate_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs) 
Handlesbtnupdate.Click
'update of flight id,name,form,to,date,hour,amount
cn.Open()
qry = "update flight_master set 
Flight_id="&txtid.Text&",Flight_name='"&txtname.Text&"',Flight_From='"&ComboBox_from.
Text&"',Flight_To='"&ComboBox_to.Text&"',Flight_Date='"&DateTimePickerdate.Text&"',Flig
ht_Hour="&txtf_hour.Text&",Flight_Amount="&txtf_amt.Text&" where Flight_Id="&txtid.Text
cmd = NewOleDbCommand(qry, cn)
cmd.ExecuteNonQuery()
cn.Close()
MsgBox("RECORD UPDATED SUCCESSFULLY....")
fillgrid()
EndSub
EndClass
```

View Booking

Description: User Flight Booking in Admin Side

Coding

```
ImportsSystem.Data.OleDb
PublicClassviewbook
'object of viewbook
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data 
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio 
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
```

```
DimqryAsString
DimadpAsNewOleDbDataAdapter
Dim ds AsNewDataSet

PrivateSubviewbook_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load
fillgrid()
EndSub

PrivateSub Button1_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handles Button1.Click
'clear
ds.Clear()
qry = "select * from book where book_id="&txtbookid.Text
adp = NewOleDbDataAdapter(qry, cn)
adp.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)
EndSub
Subfillgrid()
'fillgrid
ds.Clear()
qry = "select * from book"
adp = NewOleDbDataAdapter(qry, cn)
adp.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)


EndSub
EndClass
```

User Information

Description: User Information in Admin Side

Coding

```
ImportsSystem.Data.OleDb
PublicClassviewuser
'object of viewuser
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")

DimqryAsString

Dim dap AsNewOleDbDataAdapter
Dim ds AsNewDataSet
PrivateSubviewuser_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load
'data adapter of all data
ds.Clear()
```

```
qry = "select * from user_master"
        dap = NewOleDbDataAdapter(qry, cn)
dap.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)
EndSub


PrivateSubbtn_submit_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtn_submit.Click
ds.Clear()
qry = "select * from user_master where user_id="&txt_uid.Text
        dap = NewOleDbDataAdapter(qry, cn)
dap.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)
EndSub

PrivateSubbtn_dispall_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
ds.Clear()
qry = "select * from user_master"
        dap = NewOleDbDataAdapter(qry, cn)
dap.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)
EndSub


EndClass
```

User Login Page

Description: User Login Form

Coding

```
ImportsSystem.Data.OleDb
```

```
PublicClassuser_login
'object of user_login
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
DimdrAsOleDbDataReader

PrivateSubbtncreaccuser_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handlesbtncreaccuser.Click
'showuser_master
Dim x AsNewuser_master
x.Show()

EndSub

PrivateSubbtn_login_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtn_login.Click
'insert of first_name n password
cn.Open()
qry = "select * from user_master where User_FirstName='"&txtfname.Text&"' and
User_password="&txtpass.Text

cmd = NewOleDbCommand(qry, cn)
dr = cmd.ExecuteReader
Ifdr.HasRowsThen
dr.Read()
Dim x AsNewuserMID
x.Show()
Me.Hide()
Else
MsgBox("U HAVE TO SIGN UP THEN U CAN LOGIN...", MsgBoxStyle.Critical, "NOT
FOUND.....")

EndIf
cn.Close()
EndSub

PrivateSubuser_login_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load

EndSub
EndClass
```

Sign Up Form New User

Description: User Sign up Form (Create New User)

Coding

```
ImportsSystem.Data.OleDb
```

```
PublicClassuser_master
'object of user_master
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
Dim dap AsNewOleDbDataAdapter
Dim ds AsDataSet
PrivateSubuser_master_Load(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesMyBase.Load

EndSub
Dim gender AsString
'insert of user_master of
id,name,lastname,address,gender,DOB,age,emailid,contact,pancard,passport,adharcard,pa
ssword
PrivateSubbtn_sub_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtn_sub.Click
IfradM.CheckedThen
            gender = radM.Text
Else
            gender = radF.Text
EndIf
cn.Open()
qry = "insert into
user_mastervalues("&txtid.Text&",'"&txtfname.Text&"','"&txtname.Text&"','"&txtlname.Text&"
','"&txtadd.Text&"','"& gender
&"','"&DateTimePickerdob.Text&"',"&txtage.Text&",'"&txtmail.Text&"',"&txtcont.Text&","&txtp
an.Text&","&txtpassnum.Text&","&txtadar.Text&","&txtpass.Text&")"
cmd = NewOleDbCommand(qry, cn)
cmd.ExecuteNonQuery()
Dim x AsNewuser_login
x.Show()
cn.Close()
MsgBox("RECORD INSERTED SUCCESSFULLY...", MsgBoxStyle.Information)
EndSub
'clear
Subclear()
Dim x AsControl

ForEach x InMe.Controls
IfTypeOf x IsTextBoxThen
x.Text = ""
EndIf
Next
EndSub

PrivateSubDateTimePickerdob_ValueChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesDateTimePickerdob.ValueChanged
'age
Dim d1 = DateTimePickerdob.Value
Dim y1 = d1.Year
Dim d2 = Date.Now
Dim y2 = d2.Year

txtage.Text = y2 - y1

EndSub
EndClass
```

User Home Page

Description: User Home Page after Login

Coding

```
PrivateSubFlightDetailsToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesFlightDetailsToolStripMenuItem.Click
'object of flight_master n show
Dim x AsNewfilght_master
x.Show()
EndSub

PrivateSubUserLoginToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)
'object of user_login n show
Dim x AsNewuser_login
x.Show()
EndSub

PrivateSubUserMasterToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs)
'object of user_master n show
Dim x AsNewuser_master
x.Show()
EndSub

PrivateSubBookToolStripMenuItem_Click(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesBookToolStripMenuItem.Click
'object of book n show
Dim x AsNewbook
x.Show()
EndSub

PrivateSub ViewFlightToolStripMenuItem_Click_1(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) HandlesViewFlightToolStripMenuItem.Click
'object of viewflight n show
Dim x AsNewviewflight
x.MdiParent = Me
x.Show()
EndSub
```

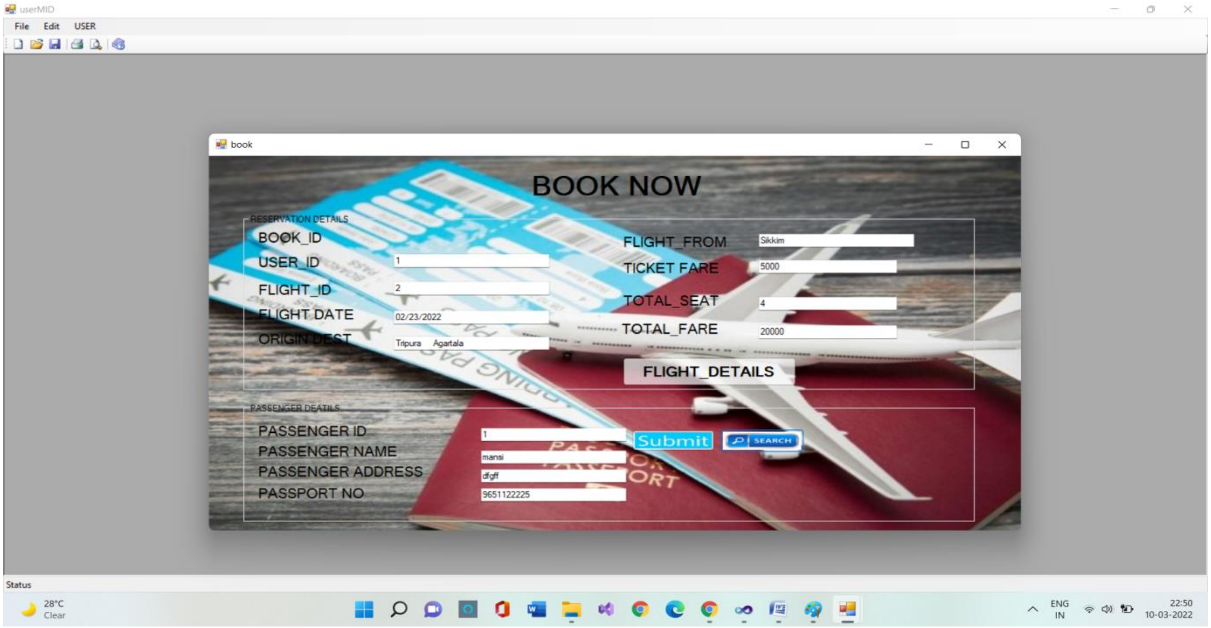
View Flight

Description: View Flight For Book

Coding

```
ImportsSystem.Data.OleDb

PublicClassviewflight
'object of viewflight
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
Dim dap AsNewOleDbDataAdapter
Dim ds AsNewDataSet
PrivateSubviewflight_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
HandlesMyBase.Load

EndSub
DimdrAsOleDbDataReader

PrivateSubbtnview_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtnview.Click
' flight_to,flight_from,flight_date
ds.Clear()
qry = "select * from flight_master where flight_to='"&comboTo.Text&"' and
flight_from='"&combofrom.Text&"'"
        dap = NewOleDbDataAdapter(qry, cn)
dap.Fill(ds)
        DataGridView1.DataSource = ds.Tables(0)

EndSub
EndClass
```

Book Now



BOOK NOW

RESERVATION DETAILS

BOOK_ID

USER_ID 1

FLIGHT_ID 2

FLIGHT DATE 02/23/2022

ORIGIN DEST Tripura Agartala

FLIGHT_FROM Sikkim

TICKET FARE 5000

TOTAL_SEAT 4

TOTAL_FARE 20000

FLIGHT_DETAILS

PASSENGER DEATILS

PASSENGER ID 1

PASSENGER NAME mansi

PASSENGER ADDRESS dfgff

PASSPORT NO 9651122225

Submit

SEARCH

Description: For Flight Booking

Coding

```
ImportsSystem.Data.OleDb
PublicClassbook
'objects of book
DimcnAsNewOleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\bs_da\OneDrive\Documents\Visual Studio
2010\Projects\airline_management\Airline.mdb")
DimcmdAsNewOleDbCommand
DimqryAsString
DimadpAsNewOleDbDataAdapter
Dim ds AsNewDataSet
DimdrAsOleDbDataReader

PrivateSubsubmit_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlessubmit.Click
'insert of book id,userid,flightid,flightdate,flight_to,flightform,ticket_fare,ticket_seat,total_fare
cn.Open()
qry = "insert into book
(user_id,flight_id,flight_date,flight_to,flight_from,ticket_fare,ticket_seat,total_fare)
values("&txtuserid.Text&","&txtfid.Text&",'"&DateTimePickerdate.Text&"','"&ComboBox_to.T
ext&"','"&ComboBox_from.Text&"',"&txtf_amt.Text&","&txt_seat.Text&","&txttotal.Text&")"
cmd = NewOleDbCommand(qry, cn)
cmd.ExecuteNonQuery()
MsgBox("RECORD INSERTED SUCCESSFULLY", MsgBoxStyle.Information)
cn.Close()
EndSub

PrivateSubbtnsearch_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtnsearch.Click
'search of book id,username,user_firstname,user_address,user_passport
Iftxtid.Text<>""Then
cn.Open()
```

```
qry = "select * from user_master where user_id="&txtid.Text
cmd = NewOleDbCommand(qry, cn)
dr = cmd.ExecuteReader
Ifdr.HasRowsThen
dr.Read()

txtid.Text = dr("user_id")
txtuserid.Text = dr("user_id")
txtfname.Text = dr("user_firstname")
txtadd.Text = dr("user_address")
txtpassnum.Text = dr("user_passport")
EndIf
cn.Close()
Else

EndIf

EndSub

PrivateSubbtn_disp_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
Handlesbtn_disp.Click
'display of ,flightid,flightdate,flight_to,flightform
Iftxtfid.Text<>""Then
cn.Open()
qry = "select * from flight_master where flight_id="&txtfid.Text
cmd = NewOleDbCommand(qry, cn)
dr = cmd.ExecuteReader
Ifdr.HasRowsThen
dr.Read()
txtfid.Text = dr("flight_id")
DateTimePickerdate.Text = dr("flight_date")
ComboBox_to.Text = dr("flight_to")
ComboBox_from.Text = dr("flight_from")
txtf_amt.Text = dr("flight_amount")

EndIf
cn.Close()
Else
MsgBox("Please Enter FLight Id")
EndIf

EndSub

PrivateSubtxt_seat_TextChanged(ByVal sender AsSystem.Object, ByVal e
AsSystem.EventArgs) Handlestxt_seat.TextChanged
'total amount
txttotal.Text = Val(txtf_amt.Text) * Val(txt_seat.Text)

EndSub
EndClass
```

Testing

4.1 Unit Testing

- A unit is the smallest testable part of software. It usually has one or a few inputsand usually a single output.
- In procedural programming a unit may be an individual program, function, procedure, etc.
- In object-oriented programming, the smallest unit is a method, which may belongto a base/super class, abstract class or derived/child class.
- Some treat a module of an application as a unit. This is to be discouraged asthere will probably be many individual units within that module.
- Unit testing is the first level of testing and is performed prior to **IntegrationTesting**.
- Unit Testing is normally performed by software developers themselves or theirpeers. In rare cases it may also be performed by independent software testers.

4.2 Functional Testing

Functional Testing is a type of software testing whereby the system is testedagainst the functional requirements/specifications.

Functions (or features) are tested by feeding them input and examining theoutput. Functional testing ensures that the requirements are properly satisfiedby the application. This type of testing is not concerned with how processingoccurs, but rather, with the results of processing.

During functional testing, **Black Box Testing** technique is used in which theinternal logic of the system being tested is not known to the tester.

- User Registration Process has been properly working and all fields arevalidated through various validation checks.
- Login and Password validation process has been co-operated properly.
- Email system module has been done properly and accurately.
- Order is being booked perfectly.
- Sending invoice through mails is working properly.
- Admin side handling and dynamic and real-time data generation module is implemented successfully.

4.3 Environment Testing

- **System Testing** is a level of the software testing where complete and integrated software is tested.
- The purpose of this test is to evaluate the system’s compliance with the specified requirements.
- The process of testing an integrated system to verify that it meets specified requirements.
- System Testing performed after **Integration Testing** and before **Acceptance Testing**.
- Normally, independent Testers perform System Testing.
- Firefox browser, Internet explorer and chrome consider testing for environment operability of software.

- Database – MS - ACCESS
- OS – Windows
- SOFTWARE – VISAUL STUDIO 2010

- Purpose of this website is to make the work easier for customers.
- Automation of the system improves the efficiency level.
- It is a friendly graphical user interface when compared to the system that already exists.
- It give authenticate access to the authorized users depending upon their user type.
- It effectively overcomes the delay in communications.
- The System has adequate scope for modification in future if it is necessary.

Bibliography & Reference

- http://www.makemytrip.com
- http://www.w3school.com
- http://www.w3layout.com
- http://www.yatra.com
- http://www.redbus.in
- http://m.youtube.com