

**A Project Report**

**‘JET AIRWAYS‘**

By

**Sonam singh**

**( B.Sc.CS )**

Submitted To

**University Of Mumbai**

Under Guidence Of

**MRS.TRIVENI KAUL**

**(2011-2012)**

**INDEX**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **TITLE** | **PAGE NO.** |
| **1** | **ACKNOWLEDGEMENT** |  |
| **2** | **COMPANY PROFILE** |  |
| **3** | **ORGANIZATIONAL OVERVIEW** |  |
| **4** | **OBJECTIVE OF THE PROJECT** |  |
| **5** | **FUNCTION OF PROJECT** |  |
| **6** | **ADVANTAGES OF PROPOSED SYSTEM** |  |
| **7** | **ENVIRONMENT USED TO DEVELOP SYSTEM** |  |
| **8** | **OPERATING ENVIRONMENT** |  |
| **9** | **DISCRIPTION OF USED SOFTWARE** |  |
| **10** | **SYSTEM ANALYSE AND DESIGN** |  |
| **11** | **SYSTEM LIFE CYCLE** |  |
| **12** | **GANTT CHART** |  |
| **13** | **SYSTEM FLOW CHART** |  |
| **14** | **ENTITY RELATIONSHIP DIAGRAM** |  |
| **15** | **CONTEXT LEVEL DIAGRAM** |  |
| **16** | **DATAFLOW DIAGRAM** |  |
| **17** | **STRUCTURE CHART** |  |
| **18** | **MENU TREE** |  |
| **19** | **DATABASE DESIGN** |  |
| **20** | **EVENT TABLE** |  |
| **21** | **PROGRAM LIST** |  |
| **22** | **REPORT LIST** |  |
| **22** | **FORMS LAYOUT** |  |
| **23** | **CONCLUSION** |  |
| **24** | **BIBLIOGRAPHY** |  |

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude towards the **Information** **Technology Department of TILAK DEGREE COLLEGE VASHI**.

It gives me the pleasure to present this project on **JET AIRWAYS** and would like to take this opportunity to thanks everyone who has been a part of making this project.

My gratitude and sincere thanks to **Mrs.** **TRIVENI KAUL** (Coordinator CS Department,TILAK DEGREE COLLAGE VASHI) for providing all the necessary system facilities and privileges and motivation that enabled a me to complete the project in time.

I would like to thank my project guide **Mrs. TRIVENI KAUL** for their throughout support and patience and guidance. I would like to thank JET AIRWAYS for allotting me the project work.

Finally I would like to thank my colleagues and friends who help me for project guidance.

**COMPANY PROFILE**

**“JET AIRWAYS”** Company was founded in 1993 **by Mr.Naresh Goyal**. Naresh Goyal (59), the founder chairman of Jet Airways.

**“JET AIRWAYS”** has come a long way since its first flight in 1993. It’s one of the fastest growing airlines in the world. It operates flights to 18 international destinations, offering a better choice in the skies.

* Company Name: JET AIRWAYS (I) PVT.LTD.
* Address: Jet Airways (I) Pvt.Ltd. , Airport Terminal, Vile Parle.
* Name of the project: JET AIRWAYS

**ORGANIZATION OVERVIEW**

**“JET AIRWAYS”** is India’s premier private airlines. Mr.Naresh Goyal is currently the chairman of Jet Airways. Jet Airways operates over 320 flights daily to 43 destinations in India and currently controls about 40% of India’s aviation market. Jet Airways was the first private airline of India to fly to international destinations. It operates daily International flights to Colombo, Katmandu, Singapore, Kuala Lumpur and Landon (Heathrow). Jet Airways has won a number of awards in recognition of standards of its service and has also received the ISO 9001:2000 certification for its In-flight Services.

Jet Airways was established on 3 May 1991 with a fleet of 4 Boeing 737-300 aircraft, with 24 daily flights serving 12 destinations. Jet Airways presently operates 55 aircrafts and is now a public limited company. Its fleet of 55 aircrafts include 3 Airbus 340-300E, 4 Boeing 737-800, 1 Airbus 320-200, 1 Boeing 737-700, 18 Boeing 737-800, 8 ATR 72-500, 2 Boeing 737-900, 12 Boeing 737-700, and 6 Boeing 737-400.

**OBJECTIVE AND SCOPE OF**

**PROJECT**

The System is designed keeping in mind following objectives:

1. To keep information of Future Flights by doing Flight Scheduling.
2. Giving Briefing Reports to crew and taking respected information of flight from crew.
3. To keep information of all Flights after takeoff and landing.
4. Keep the record of Fuel Management by doing Scheduling.
5. To keep the record of number of passenger in each Flight.
6. To keep the information of all Aircraft in JET AIRWAYS including their Aircraft Specification and Seat Map View.
7. Keep record of Employee with their personal detail and attendance record.
8. Keeping record of Events, Meetings, Training held by JET AIRWAYS.
9. Multi-User Environments.

10) Rights to the users.

11) Warnings if wrong information is filled.

12) Reports and other informatory reports.

1. **To keep information of Future Flights by doing Flight Scheduling**.

By doing Flight user can arrange the Flight including all information about Flight, time and information of crew & cabin.

1. **Giving Briefing Reports to crew and taking respected information of flight from crew**.

This report will contain all information about Flight with any changes.

1. **To keep information of all Flights after takeoff and landing**.

This information provides all the details of Flight after taking takeoff and landing. It includes schedule time, AON, AOF, delay time.

1. **Keep the record of Fuel Management by doing Scheduling**.

By doing Fuel Management user can manage the Fuel for each Flight.

1. **To keep the record of number of passenger in each Flight.**

This data only stores the no. of passengers in each flight.

1. **To keep the information of all Aircraft in JET AIRWAYS including their Aircraft Specification and Seat Map View**.

This keeps the information of all Aircraft including their name, quantity, no. of seats and also their seat map view.

1. **Keep record of Employee with their personal detail and attendance record**.

All the employee details are stored with their Employee id, Name, Address, Contact no. , Email-id etc. Attendance record will contain employee id and their respective time-in, time-out along with their over-time and leaves.

1. **Keeping record of Events, Meetings, Training held by JET AIRWAYS**.

The record contain all information about Events, Training, Meeting organized by JET AIRWAYS.

**9) Multi-User Environments**.

A multi-user environment enables the system to be handled by many users. Thus the system works efficiently even in the absence of the Authorized person in an emergency. Hence the information cannot be misused by unauthorized person. Also a check as been kept on the system so that authorized person comes to know who has added, modified and deleted particular records of , employee & passengers by which ID and at what time i.e. login\_ date,

login\_ time and logout\_ time in the database.

**10)** **Rights to the users**.

Different rights have been assigned to the user as per the needs of the organization. The user can perform tasks like adding, updating and deleting record.

1. **Warnings if wrong information is filled**.

It gives warnings and error message if the expected information is not filled.

1. **Reports and other informatory reports**.

This gives the report & information to the other department.

**FUNCTION OF THE PROJECT**

This system performs the following functions:

**Flight Schedule**:

Enables the user to make Schedules of Flight which is then stored in the database. This also provides facility of details of Flight Schedules so the user comes to know which flight scheduling is done before. This entry can be later used for retrieving for making flight data report.

**Fuel Management:**

In fuel management, user can do the scheduling of fuel. Once the flight scheduling is done, fuel department manages the fuel quantity for each flight.

**Flight Entry:**

This enables the user to fill the information of flight after the flight landing to its Destination. This gives the details of each flight including their ATD, AOF, and AON timings. Also provides the information of the flight if the flight is Delay with their respective delay time and Reason for the Delay.

**Dispatch Briefing**:

In dispatch briefing, the flight dispatch officer gives the briefing reports to the crew. The report contains information about the details of flight including fuel info, passenger’s info, alternate destination, weather info and etc.

**Crew Briefing:**

In crew briefing, the crew gives the report to the flight dispatch officer. The reports contain information of flight including actual time info, fuel info, crew, cabin crew info and etc.

**Fleet**:

This provides information of all aircraft currently used by JET AIRWAYS with aircraft specification including aircraft name engine speed, cruise speed, aircraft length wing span and seat configuration.

**Meeting, Training and Events**:

This provide the information about the meeting, training, events organized by JET AIRWAYS including their date, time etc.

**Employee Designation**:

This enables user to make entry of designation as per the department. And according to department and designation employees gross salary, pay-leaves and etc. will be decided.

**Employee Entry:**

In employee entry, user can enter the personal details of new employee such as Name, Address, Qualifications, Contacts, etc as well as company details such as departments, designations, date of joining, status etc. While doing employee entry the Branch Manager allots the password to each employee which is then used for making attendance of employee.

**Employee Attendance:**

Using the Employee ID and password the employee enters his/her attendance. According to attendance the salary will be decided.

**Employee Pay-Slip:**

In the Payment slip of Employee the Net Salary per month is calculated and report is generated.

**ADVANTAGES OF PROPOSED SYSTEM**

1. **Direct access to Airline-partner website :**

The system provides the direct access to the Airline Partner website. User can access whenever he wants during doing his normal work. No need to open Internet Explorer instead by just clicking on name of company he will get that website.

1. **Providing Weather Link :**

User can access the weather website for entering the weather information, and can easily watch all country weather anytime.

1. **Providing Contacts of Employee :**

The system provides the details of all employees working in JET AIRWAYS.

1. **Showing the Aircraft Seat mapping :**

The system can show the images of Aircraft Seat mapping to the employee so user comes to know about capacity of Aircraft by just seeing the image.

1. **Good Search Facility :**

User can search Flight info into by just entering no and other related information in every few minutes.

1. **Providing Password For Attendance :**

In a system, admin can give a password to each employee at the time of employee entry, and this password will be used by employee for making their attendance.

1. **Editing Attendance :**

User have rights to change the details of employee attendance in case of attendance of employee is not marked.

1. **Reduction of Paper Work :**

Maintenance of all information in database files is there by reducing manual work &consecutively reducing the hue volume of tedious paper work.

1. **Reports Generation for Management :**

The system provides various reports that will help to keep the records of information.

**ENVIRONMENT USED**

**TO DEVELOP SYSTEM**

**Hardware**:

* Windows Operating System.
* 1 GB RAM or more.
* Pentium IV Intel and above.
* 160 GB Hard Disk.

**Software:**

* Frontend: Microsoft Visual Basic 2008 Express Edition.
* Backend: Microsoft SQL Server.
* Adobe Photoshop 7.0
* Internet

**OPERATING ENVIRONMENT**

The minimum configuration required for the software to work is as follows

**Hardware details:**

* P-III 600 MHz
* 64 MB RAM
* 40 GB HDD
* 1.44 MB FDD

**Software Details:**

* Microsoft Visual Basic 2008 Express Edition
* Microsoft SQL Server
* Microsoft Operating System 98 or XP above.

Above given is the minimum configuration required for efficient working of the software can run with more speed on platform such as the P-IV processor, with Windows XP operating system.

**Visual Studio 2008**:-

**Microsoft Visual Studio** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](http://en.wikipedia.org/wiki/Microsoft). It can be used to develop [console](http://en.wikipedia.org/wiki/Console_application) and [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) [applications](http://en.wikipedia.org/wiki/Application_software) along with [Windows Forms](http://en.wikipedia.org/wiki/Windows_Forms) applications, [web sites](http://en.wikipedia.org/wiki/Web_site), [web applications](http://en.wikipedia.org/wiki/Web_application), and [web services](http://en.wikipedia.org/wiki/Web_service) in both [native code](http://en.wikipedia.org/wiki/Native_code) together with [managed code](http://en.wikipedia.org/wiki/Managed_code) for all platforms supported by [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [Windows Mobile](http://en.wikipedia.org/wiki/Windows_Mobile), [Windows CE](http://en.wikipedia.org/wiki/Windows_CE), [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework), [.NET Compact Framework](http://en.wikipedia.org/wiki/.NET_Compact_Framework) and [Microsoft Silverlight](http://en.wikipedia.org/wiki/Microsoft_Silverlight).

Visual Studio includes a [code editor](http://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](http://en.wikipedia.org/wiki/IntelliSense) as well as [code refactoring](http://en.wikipedia.org/wiki/Code_refactoring). The integrated [debugger](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building [GUI](http://en.wikipedia.org/wiki/GUI) applications, web designer, [class](http://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](http://en.wikipedia.org/wiki/Database_schema) designer.

It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source-control](http://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](http://en.wikipedia.org/wiki/Subversion_(software)) and [Visual SourceSafe](http://en.wikipedia.org/wiki/Visual_SourceSafe)) and adding new toolsets like editors and visual designers for [domain-specific languages](http://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](http://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](http://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

Visual Studio supports different [programming languages](http://en.wikipedia.org/wiki/Programming_language) by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any [programming language](http://en.wikipedia.org/wiki/Programming_language), provided a language-specific service exists. Built-in languages include [C](http://en.wikipedia.org/wiki/C_(programming_language))/[C++](http://en.wikipedia.org/wiki/C%2B%2B) (via [Visual C++](http://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](http://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](http://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (via [Visual C#](http://en.wikipedia.org/wiki/Visual_C_Sharp)), and [F#](http://en.wikipedia.org/wiki/F_Sharp_(programming_language)) (as of Visual Studio 2010[[2]](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-1)). Support for other languages such as [M](http://en.wikipedia.org/wiki/M_(programming_language)), [Python](http://en.wikipedia.org/wiki/IronPython), and [Ruby](http://en.wikipedia.org/wiki/IronRuby) among others is available via language services installed separately. It also supports [XML](http://en.wikipedia.org/wiki/XML)/[XSLT](http://en.wikipedia.org/wiki/XSLT), [HTML](http://en.wikipedia.org/wiki/HTML)/[XHTML](http://en.wikipedia.org/wiki/XHTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript) and [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets). Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

Microsoft provides "Express" editions of its Visual Studio 2010 components Visual Basic, Visual C#, Visual C++, and Visual Web Developer at no cost.

Visual Studio 2010, 2008 and 2005 Professional Editions, along with language-specific versions (Visual Basic, C++, C#, J#) of Visual Studio 2005 are available for free to students as downloads via Microsoft's [DreamSpark](http://en.wikipedia.org/wiki/DreamSpark) program.

Visual Basic.NET is the most recent generation of Visual Basic. Developers will be pleased to note that its new features include inheritance, method overloading, structured exception handling, and more. These capabilities make it easier than ever to create .NET applications, including Windows applications, web services, and web applications

**SYSTEM ANALYSIS AND DESIGN**

1. **Feasibility study and problem definition:**

In this phase the problem is defined by and the technical feasibility, economic feasibility and operational/ behavioral feasibility is measured. Feasibility refers to the ability of the company to handle the various aspects of the company regarding the operations which is undertakes.

The techniques which it implements while working and finances required in order to cope-up with the project. If the company is operationally secured to accept the project then it is said that the firm is operationally feasible.

1. **Requirement Analysis and Specification:**

Requirement analysis provides detailed study of the requirement of both the user and the software. This activity is basically concerned with the purpose of the system for which it is being made and analyzes the parameters such as:

* Input to the system
* Process required
* Output expected
* Contains

1. **Design and specification:**

This phase deals with various concepts of the system design such as data structure, software architecture, and algorithms. In this phase we translate the requirement into a representation of the software.

Designing basically deals with the designing of the entire software framework. At this stage various requirements as per the technical point of view are started so that the design works out to be efficient and acceptable to the user.

1. **Coding and Module Testing:**

In this phase the actual coding of the software is done and each module is tested to work as desired. Coding refers to the translation of the design in to machine readable form. The more detailed the design is, the path for coding becomes much clear and thus coding can be implemented with better reliability.

1. **Integration and system Testing:**

Once the coding is done, each module is written, it’s the perfect time to integrate all modules and all the modules are tested for their correctness and accuracy. Interoperability of various modules is also looked after. Testing requires a detailed plan as to what, when and how to test.

Testing basically deals with taking care of customer’s expected results and the desired accuracy as per the customer.

1. **Delivery(Implementation):**

After integration and testing, when software is found to be correct it is needed to be implemented at the user’s side. Delivery phase include training of the operations associated with the software to the users of the system.

1. **Maintenance:**

After the implementation of the project it is evaluated to work satisfactorily to perform the entire task desired from it. If any further requirement arises, it should be maintained and if any corrections are needed that should also be done in this phase.

1. **Fact Finding Technique:**

In system analysis we have to find a number of fact finding techniques about the system which is under study. These facts help us to understand, what is the existing system, the flow of the existing system and what is the problem in the existing system.

**SYSTEM LIFE CYCLE**

The system life cycle phases:

The Waterfall Model also called the **Linear Sequential Model** or **Classic** **Life Cycle Model**, is the traditional life cycle model, where each phases has a defined a star point and an end point, and clear deliverables from one phase to the next. The sequential moves from one place to the next gave rise to the name Waterfall Model. This model is useful because of the specific goal at the end of phase, rather only at the end of the

**Requirement Analysis**

**System Engineering**

**System Design**

**Coding**

**Testing**

**Maintenance**

Fig: System Life Cycle

* **System/ Information Engineering and modeling:**

Because software I always a part of a larger system (or business). The work begins by establishing requirements for all system and then allocating some of these requirements to the software. This system view is essential when software must interface with other elements such as hardware, people, and databases.

System engineering and analysis encompasses requirements gathering at the system level with a small amount of the top level analysis and design. Information engineering encompasses requirements gathering at the strategic business level and at the business area level.

* **Software requirement analysis:**

The requirement gathering process is intensified and focused specifically software engineer (analysis) must understand the information domain for the software as well as the required function, behavior, performance and interfacing. Requirements for both, the system and the software are documented and reviewed by the customer.

* **Design:**

Software design is actually a multi-step process that focuses on four distinct attributes of a program: data structure, software, architecture, interface representation, and procedural detail.

The design process translates the requirements into representation of the software that can be assessed for quality before the code generation begins. Like requirements the design is also documented and becomes part of the software configuration.

* **Code generation:**

The design mustbe translatedinto a machine readable form. The code generation step performs this task. If the designing is done on a detailed manner, code generation can be achieved mechanically.

* **Testing:**

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software assuring that all statements have been tested and the functional external i.e. conducting tests to uncover error and ensure that defined input will produce actual results that agree with the desired or the requires results.

* **Maintenance:**

Software will undoubtedly undergo changes after it is delivered to the customer. Change will occur because error may have been encountered, software must adapt to the changes in the external environment; or customer requires some functional or performance enhancements. Software maintenance replies each of the preceding phases to an existing program rather than a new one.

The sequential model provides a template into which methods for analysis, design, coding, testing, and maintenance can be placed. The classical life cycle remains the most widely used process model for software engineering. While it does have weakness, it is significantly better than a haphazard approach to software development.

The system life cycle consists of three parts namely:

1. Context Level Diagram
2. Entity Relationship Diagram(ERD)
3. Dataflow Diagram(DFD)

**Gantt Chart**

A Gantt chart is a project management tool used to schedule, organize and coordinate tasks within a project. The Gantt chart is named after Charles Gantt, and is not an acronym. It is a schedule (time ordered listing) of planned events.

* During the area of scientific management, Henry Gantt developed a tool for displaying the progression of a project in the form of a specific chart. Gantt’s scheduling tool takes the form of a horizontal bar graph and is known as Gantt chart.
* The horizontal axis of the Gantt chart is a time scale, expressed either in absolute time referenced to the beginning of the project. The time resolution depends on the project, the time unit typically is weeks or months.
* For larger projects, a work breakdown structure would be developed to identify the tasks before constructing a Gantt chart. For smaller projects, the charts itself may used to identify the tasks.

* The strength of this chart is its ability to display the status of each activity at a glance. For sequencing and critical path analysis, network models such as CPM and PERT are more powerful for dealing with dependencies and project completion time. Even when network models are used, the Gantt chart is often used as reporting tools.

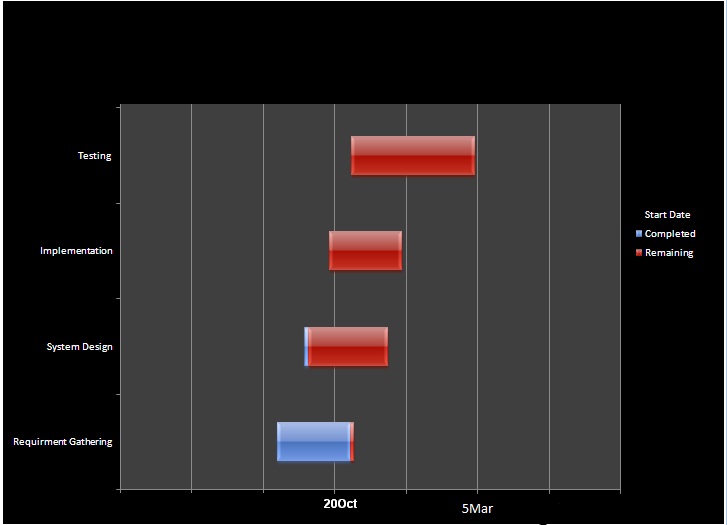
**Gantt charts are useful tools for planning and scheduling projects.**

1. Gantt charts allow you to access how long a project should take.
2. Gantt charts lay out the order in which tasks need to be carried out.
3. Gantt charts help manages the dependencies between tasks.
4. Gantt charts determine the resources needed.

**Gantt charts are useful tools when a project is under way.**

1. Gantt charts monitor progress. You can immediately see what should have been achieved at a point in time.
2. Gantt charts allow you to how remedial action may bring the project back on course.

The Gantt chart is useful way to present the general flow of project’s tasks. Such charts are particularly useful for coordinating multiple activities. The Gantt chart also provides a useful way to track project progress against the plan and schedule.



**SYSTEM FLOW CHART**

Details of Flight

Details of Flight Schedule

**Flight Schedule**

Details of

Passengers

Details of

Flight Entry

1.5

Flight Entry

Passengers

Fuel Schedule

Flight Schedule

**Fuel Schedule**

DBrief

**Passengers**

Details of Dispatch Briefing

**Dispatch Briefing**

**Flight Entry**

Fleet

Details of Fleet

Details of Event

mm

**Event**

**Fleet**

Event

Designation

**Designation**

Details of Designation

Designationdd

Details of Employee Entry

Employee Entry

**Employee Entry**

Pay Slip

Edit Attendance

Attendance

Pay Slip

Passengers

Dispatch Briefing

Fight Schedule

Fuel Schedule

Crew Briefing

Attendance

**Report Generation**

**Attendance Details**

**Edit Attendance**

**Pay Slip**

Details of Payment

Details of Edit Attendance

Details of Attendance

**ENTITY RELATIONSHIP DIAGRAM**

* **Entity-Relationship Diagram** was originally proposed by Peter Chen.
* For the design of relational database system and has been extended by others.
* Entity Relationship Diagram (ERD) provides the analyst a concise notation for examining data within the context of a software application.
* Entity Relationship Diagram (ERD) enables a software engineer to fully specify the data objects that are Input Source and Output Source from a system, the attributes that define the properties of these objects, and their relationships.
* A set of primary components are identified for the ERD:
* Data Objects
* Attributes
* Relationship
* Various type indicators

* The primary purpose of ERD is to represent data objects and their relationship.
* Connection between data objects and relationships are established using a variety of special symbols that indicate Cardinality and Modality.
* Cardinality helps us understand as to how many occurrences of another object.
* Cardinality is the specification of the number of occurrences on one (object) that can be related to the number of occurrences of another (object).

DBrief

Fuel Data

Fuel Management

Sch-Data

Sch-Data

Sch-Data

Flight Schedule

Flight Entry

Fleet

Event

SalaryData

Employee Entry

Designation

Emp & Salary Data

PaySlip

**CONTEXT LEVEL DIAGRAM**

Fuel Schedule

Schedule of Fuel Fuel schedule details

Passenger

Flight Schedule

Passengers Details

Flight Schedule

Dispatch

Briefing

Flight Schedule Details Fuel schedule

Passenger

Information of Fleet Weather

HR

Schedule info

Flight Entry

Info of Event

Info of meeting Flight time

Info of training

Attendance Briefing

Crew

Briefing

Attendance

Designation

Flight schedule Employee Entry

Admin

Report

Fuel management Pay Slip

Weather

Dispatch Briefing

**DATE FLOW DIAGRAM**

**(DFD)**

An information moves through, its modify by a series of transformation. A DFD is graphical technique that depicts internal flow and the data moves from Input Source to Output Source.

A level 0 DFD also called a Fundamental System Model or Context Model represents entire software element, with Input Source and Output Source data indicated by incoming and outgoing respectively.

Additional process and information flow paths are represented as the level 0DFD is partitioned to reveal more details. Each of the process represented as level 1 as Sub Function of overall system in the Context Model.

**Notation To Create DFD:**

Process : **Main Process**

Entity : **Small Process**

Data flow : **The direction of arrow**

**indicate the flow of data.**

Database: **The data is stores in the database**.

Level 0:

Login

User

Failed

JET

AIRWAYS

**DFD Level I:**

Attendance

1.10

Attendance

Login

1.9

PaySlip

1.1

Flight schedule

PaySlip

1.8

Event

Event

1.2

Flight

Flight Schedule

1.7

Employee entry

1.3

Fuel mgmt

Flight Entry

Employee Entry

1.6

Designation

1.4

Fleet

Fuel mgmt

1.5

Dis Briefing

Designation

Fleet

DBrief

**DFD Level 2:**

**Flight Schedule:**

Flight Schedule

2.0

2.0

Flight Schedule

Search Flight Date, Flight No

Flight Schedule Report

**Fuel Schedule:**

Fuel Schedule

2.1

2.1

Fuel Management

Search by Flight Date, Flight No

Fuel

Schedule

Report

**Flight Entry:**

Flight Entry

2.2

Flight Schedule

2

Search by Flight Date, Flight No

Flight Schedule Report

**Dispatch Briefing:**

Dispatch

Briefing

2.3

DBrief

Dispatch Briefing Report

Weather

**Fleet:**

Fleet

2.6

Fleet

**Event:**

Event

Event

2.7

**Designation:**

Designation

2.8

Designation

**Employee Entry:**

Employee Entry

2.9

Employee

Search by Emp ID

**STRUCTURE CHART**

**JET AIRWAYS**

Events

Aircraft

Employee

Fuel Management

Operation

Reports

**Attendance of Emp Aircraft Data Events**

Events

Flight Sche-dule

**Search flight**

Fuel Management

Edit Attendance

Search for Flight

Aircraft Data

Attendance

DBrief

Pay Slip

**Fuel entry weather info DBrief Schedule**

Weather

Weather

Flight Schedule

Dispatch Briefing

Fuel Manage-ment

**PaySlip**

Pay slip

**Designation New**

Designation

**Emp**

**Entry New User**

New User

New Emp Entry

**MENU TREE**

1. **Operation:**

**Attendance**

**Edit Attendance**

1. **Fuel Management:**

**Fuel Schedule**

**Fuel**

**Fuel Schedule Details**

**Weather**

1. **Aircraft:**

**Aircraft Data**

**Dispatch Briefing**

**Flight Schedule**

**Search for Particular Flight**

1. **Employee:**

**Designation**

**New Employee entry**

**New User**

**Payment Slip**

1. **Events:**

**Events**

1. **Reports:**

**Flight Schedule Report**

**Dispatch Briefing Report**

**Weather Report**

**Payment Slip Report**

**Fuel Management Report**

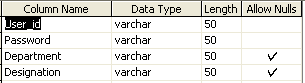
**Database Design**

Database is the key factor in any Front End Application. Database is also called as Back End. We are using Microsoft SQL server as Back End. Database allows you to create complex queries such as Make new table Queries, Add , Update, Delete the records. User Defined function etc. It allows you to use aggregate functions. The Main advantage is that using Primary Key and Foreign Key relationship you can able to Join multiple tables. SQL Server can import or export data. The database can be encrypted easily to anyone reading the data.

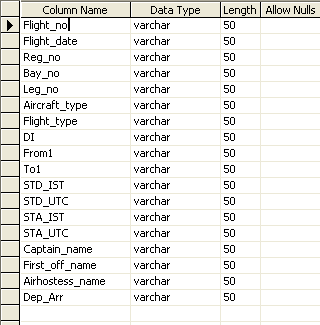
**Jet Airways Project Contain following Tables in the database**:

1. **Login**
2. **Flight Schedule**
3. **Fuel Management**
4. **DBrief**
5. **Fleet**
6. **Event**
7. **Designation**
8. **Employee Entry**
9. **Attendance**
10. **Pay Slip**

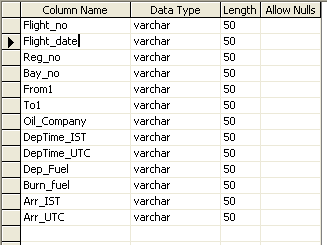
* **Login Table:**



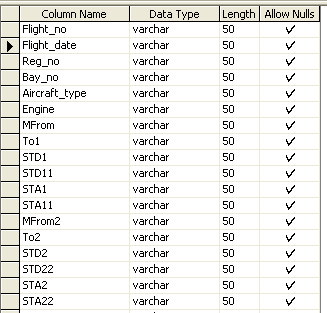
* **Flight Schedule:**

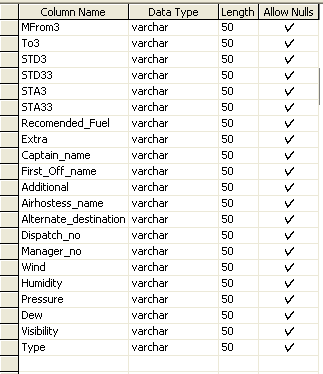
****

* **Fuel Management:**

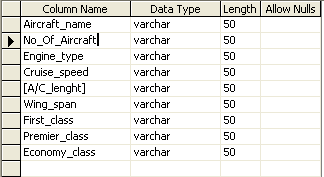
****

* **DBrief:**

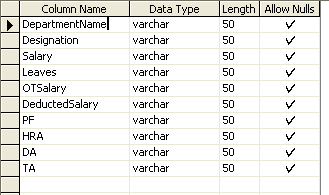
****

****

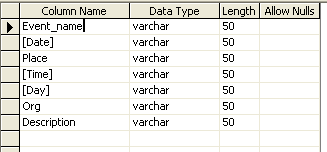
* **Fleet:**

****

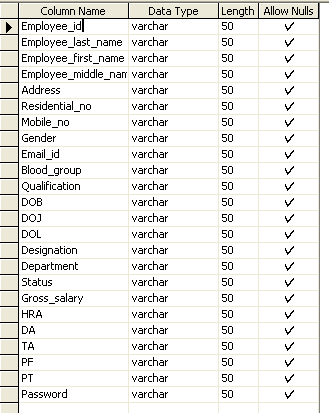
* **Designation:**

****

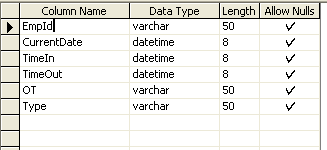
* **Events:**

****

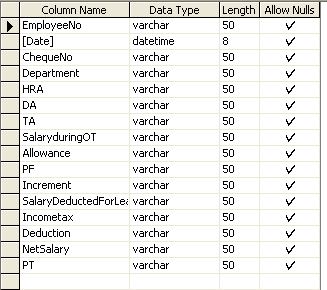
* **New Employee Entry:**



* **Attendance:**

****

* **Payment Slip:**



**EVENT TABLE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Event** | **Trigger** | **Source** | **Activity** | **Response** | **Destination** |
| **1.** | Accept Flight Schedule Entry | Regular /Daily | Flight Schedule | Save, Update, Delete the Entry | Gives the Details | Flight Operation Officer |
| **2.** | Accept Flight Entry | Regular /Daily | Flight Entry | Save the Entry | Gives the Details | Flight Operation Officer |
| **3.** | Accept Fuel Schedule Entry | Regular /Daily | Fuel Mgmt | Save, Update, Delete the Entry | Gives the Details | Fuel Mgmt Officer |
| **4.** | Accept Fuel Entry | Regular /Daily | Fuel | Save the Entry | Gives the Details | Fuel Mgmt Officer |
| **5.** | Accept Dispatch Briefing | Regular /Daily | DBrief | Save, Update the Entry | Gives the Details | Flight Dispatch Officer |
| **6.** | Arrange Events | Monthly | Events | Save, Update, Delete the Entry | Gives the Details | HR Officer |
| **7.** | Accept Employee Entry | Required | Employee Entry | Save, Update, Delete | Gives the Details | Admin |
| **8.** | Accept Attendance | Every Shift | Attendance | Save, Update the Entry | Gives the Details | Admin |
| **9.** | Make the Pay Slip | Monthly | Pay Slip | Save the Entry | Gives the Details | Admin |
| **10.** | Flight Schedule Report | Daily | Flight Schedule | Show the Report | Print the Report | Flight Operation Officer |
| **11.** | Dispatch Briefing Report | Daily | DBrief | Show the Report | Print the Report | Flight Dispatch Officer |
| **12.** | Weather Report | Daily | DBrief | Show the Report | Print the Report | Flight Dispatch Officer |
| **13.** | Payment Report | Daily | Pay Slip | Show the Report | Print the Report | Admin |
| **14** | Fuel Mgmt | Daily | Fuel Mgmt | Show the Report | Print the Report | Flight Dispatch Officer |

**PROGRAM LIST**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **Form Name** | **Description** |
| **1.** | **frmsplashscreen1.vb** | **It helps to load the project.** |
| **2.** | **frm splashscreen2.vb** | **It helps to load the project.** |
| **3.** | **frmlogin.vb** | **It helps to login the project.** |
| **4.** | **frmnewuser.vb** | **It helps to new user to login.** |
| **5.** | **frmMDI.vb** | **This form displays all the menus through which we can access other forms.** |
| **6.** | **frmflightschedule.vb** | **It gives the information about Flight Schedule.** |
| **7.** | **frmfuelmanagement.vb** | **It gives information about Fuel management.** |
| **8.** | **frmdispatchbriefing.vb** | **It gives information about Dispatch Briefing.** |
| **9.** | **frmweather.vb** | **It gives the information about weather.** |
| **10.** | **frmaircraftinformation.vb** | **It gives information about Aircraft.** |
| **11.** | **frmjetairwaysevents.vb** | **It gives the information Events.** |
| **12.** | **frmdesignation.vb** | **It gives information about Designation.** |
| **13.** | **frmnewemployeeentry.vb** | **It Provides the information about Kitchen New Employee Entry .** |
| **14.** | **frmattendance.vb** | **It Provides the information about Attendance.** |
| **15.** | **frmeditattendance.vb** | **It Provides the information about Overtime of Employee.** |
| **16.** | **frmsearchflightentry.vb** | **It helps to search particular Flight Entry Information.** |
| **17.** | **frmpamentslip.vb** | **It gives the information about payment slip of Employee.** |

**REPORT LIST**

1. **Flight Schedule Report :**

* This report gives the all details of the flight schedule.

1. **Dispatch Briefing Report:**

* This report gives the details of Flight. This dispatch officer gives this report to crew before the flight takeoff.

1. **Weather Report :**

* This report gives the details of weather of flight.

1. **Payment Report :**

* This report gives the pay slip to the employee per month.

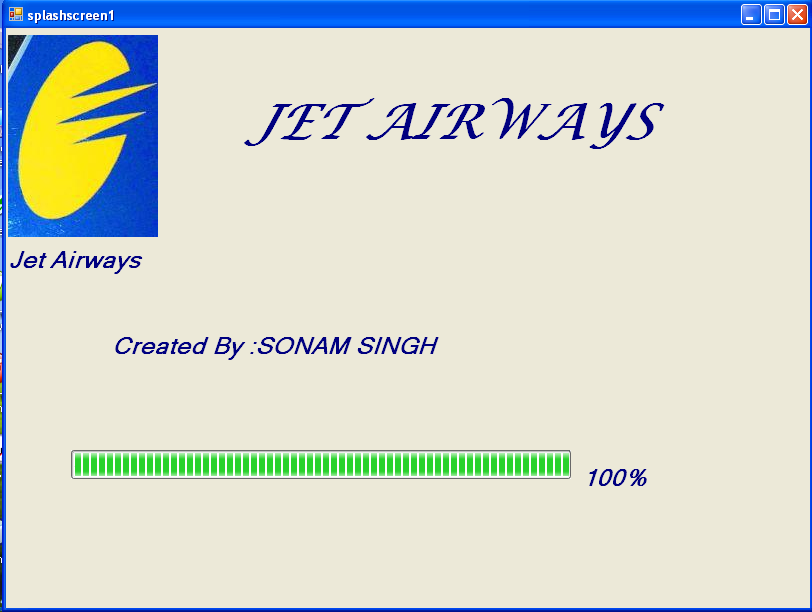
1. **Fuel Management Report :**

* This report gives the details of Fuel Management. This dispatch officer gives this report to crew before the flight takeoff.

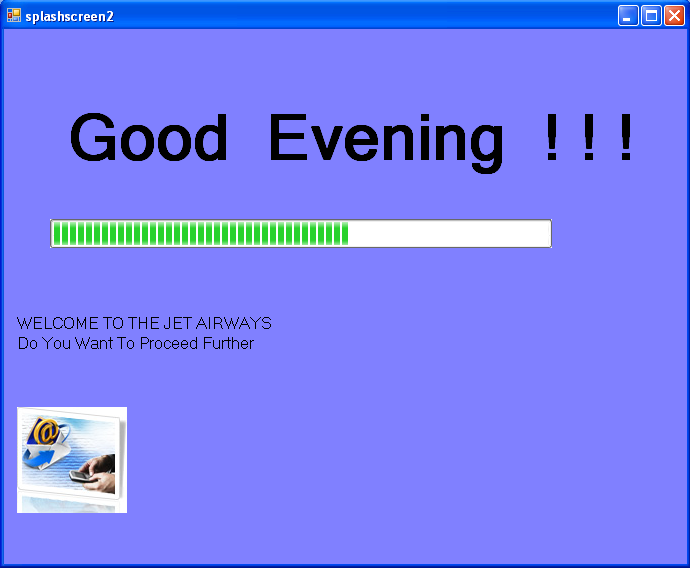
**FORMS**

**Snapshots**

**Splash Screen1**



**Splash Screen2**



**Login**



**Coding:**

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmlogin

Private Sub frmlogin\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim cmdid As New SqlCommand("select \* from Logintable", cn)

cn.Open()

Dim drid As SqlDataReader = cmdid.ExecuteReader

While drid.Read

cbouserid.Items.Add(drid.GetString(0))

End While

drid.Close()

cn.Close()

End Sub

Private Sub cbouserid\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbouserid.SelectedIndexChanged

Dim cmdfill As New SqlCommand("select \* from Logintable where User\_id='" & cbouserid.Text & "'", cn)

cn.Open()

Dim drfill As SqlDataReader = cmdfill.ExecuteReader

While drfill.Read

txtdepartment.Text = drfill.GetString(2)

txtdesignation.Text = drfill.GetString(3)

End While

drfill.Close()

cn.Close()

End Sub

Private Sub btnok\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click

Dim cmdok As New SqlCommand("select \* from Logintable where User\_id='" & cbouserid.Text & "' and Password='" & txtpassword.Text & "'", cn)

cn.Open()

Dim drok As SqlDataReader = cmdok.ExecuteReader

If drok.Read Then

Me.Hide()

MsgBox("Login Succeded....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "WELCOME")

MDIParent1.Show()

Else

MsgBox("ACCESS DENIED", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

txtpassword.Text = ""

txtpassword.Focus()

End If

drok.Close()

cn.Close()

End Sub

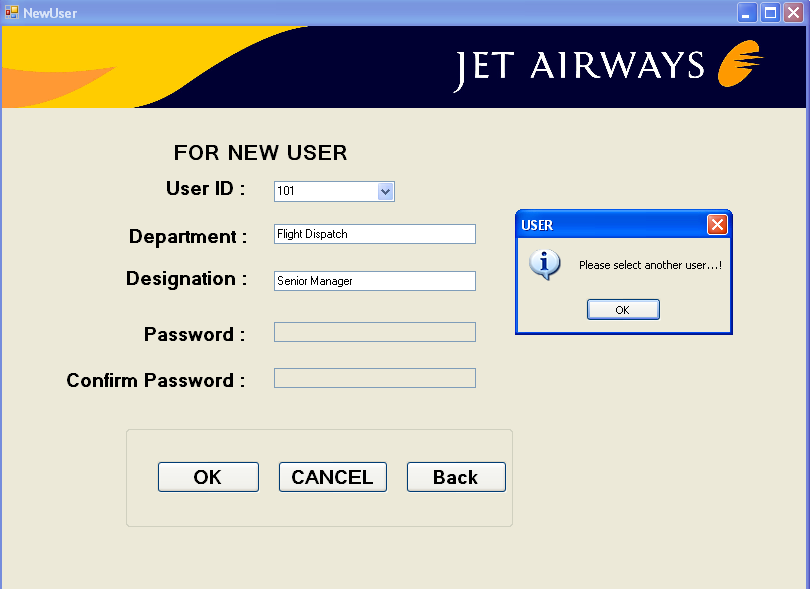
Private Sub btncancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btncancel.Click

End

End Sub

End Class

**NEW USER LOGIN**

****

**Coding:**

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmnewuser

Private Sub frmnewuser\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

txtpassword.Enabled = False

txtconfirmpassword.Enabled = False

Dim cmdid As New SqlCommand("select \* from Employee\_entry", cn)

cn.Open()

Dim drid As SqlDataReader = cmdid.ExecuteReader

While drid.Read

cbouserid.Items.Add(drid.GetString(0))

End While

drid.Close()

cn.Close()

End Sub

Private Sub cbouserid\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbouserid.SelectedIndexChanged

Dim i As Integer

Dim a As String

Dim cmdfill As New SqlCommand("select \* from Employee\_entry where Employee\_id='" & cbouserid.Text & "'", cn)

cn.Open()

Dim drfill As SqlDataReader = cmdfill.ExecuteReader

While drfill.Read

txtdepartment.Text = drfill.GetString(15)

txtdesignation.Text = drfill.GetString(14)

End While

drfill.Close()

cn.Close()

Dim cmduser As New SqlCommand("select \* from Logintable where User\_id='" & cbouserid.Text & "'", cn)

cn.Open()

Dim druser As SqlDataReader = cmduser.ExecuteReader

If druser.Read Then

MsgBox("The selected user is already assigned a password...!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "PASSWORD")

i = MsgBox("Do you want to change the password?", MsgBoxStyle.YesNo + MsgBoxStyle.Question, "PASSWORD")

If i = MsgBoxResult.No Then

MsgBox("Please select another user...!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "USER")

cbouserid.Text = ""

txtdepartment.Text = ""

txtdesignation.Text = ""

Else

txtoldpass.Text = druser.GetString(1)

a = InputBox("Please enter old password")

If a = txtoldpass.Text Then

txtpassword.Enabled = True

txtconfirmpassword.Enabled = True

Else

MsgBox("Please cheack the old password...!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

cbouserid.Text = ""

txtdepartment.Text = ""

txtdesignation.Text = ""

End If

End If

End If

druser.Close()

cn.Close()

End Sub

Private Sub btnok\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click

Dim cmdchk As New SqlCommand("select \* from Logintable", cn)

cn.Open()

Dim drchk As SqlDataReader = cmdchk.ExecuteReader

If drchk.Read Then

If cn.State = ConnectionState.Open Then

cn.Close()

End If

Dim cmduplogin As New SqlCommand("update Logintable set Password='" & txtconfirmpassword.Text & "' where User\_id='" & cbouserid.Text & "'", cn)

cn.Open()

cmduplogin.ExecuteNonQuery()

MsgBox("Record Saved successfully")

cn.Close()

Else

If cn.State = ConnectionState.Open Then

cn.Close()

End If

Dim cmdsave As New SqlCommand("insert into Logintable values('" & cbouserid.Text & "','" & txtconfirmpassword.Text & "','" & txtdepartment.Text & "','" & txtdesignation.Text & "')", cn)

cn.Open()

cmdsave.ExecuteNonQuery()

MsgBox("Record Saved successfully")

cn.Close()

End If

drchk.Close()

cn.Close()

Dim cmdup As New SqlCommand("update Employee\_entry set Password='" & txtconfirmpassword.Text & "' where Employee\_id='" & cbouserid.Text & "'", cn)

cn.Open()

cmdup.ExecuteNonQuery()

cn.Close()

End Sub

Private Sub txtconfirmpassword\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtconfirmpassword.LostFocus

If txtconfirmpassword.Text <> txtpassword.Text Then

MsgBox("Pls enter same password as above....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

txtconfirmpassword.Text = ""

txtconfirmpassword.Focus()

End If

End Sub

Private Sub btncancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btncancel.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

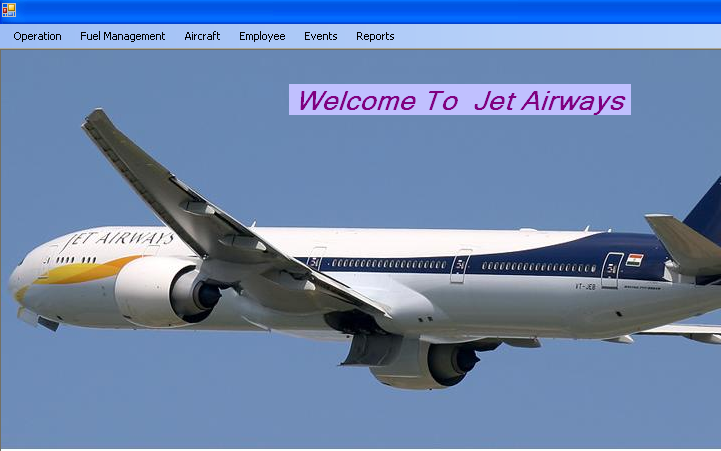
MDIParent1.Show()

Me.Hide()

End Sub

End Class

**MAIN FORM**



**Coding:**

Imports System.Windows.Forms

Public Class MDIParent1

Private Sub EditAttendanceToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.Hide()

frmeditattendance.Show()

End Sub

Private Sub FrmfuelmanagementToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmfuelmanagementToolStripMenuItem.Click

frmfuelmanagement.Show()

Me.Hide()

End Sub

Private Sub FrmaircraftdataToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmaircraftdataToolStripMenuItem.Click

frmaircraftdata.Show()

Me.Hide()

End Sub

Private Sub FrmdespatchBriefingToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmdespatchBriefingToolStripMenuItem.Click

frmdispatchbriefing.Show()

Me.Hide()

End Sub

Private Sub FrmflightscheduleToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmflightscheduleToolStripMenuItem.Click

frmflightshedule.Show()

Me.Hide()

End Sub

Private Sub FrmsearchflightentryToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmsearchflightentryToolStripMenuItem.Click

frmsearchflightentry.Show()

Me.Hide()

End Sub

Private Sub FrmdesignationToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmdesignationToolStripMenuItem.Click

frmdesignation.Show()

Me.Hide()

End Sub

Private Sub FrmnewempentryToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmnewempentryToolStripMenuItem.Click

frmnewemployeeentry.Show()

Me.Hide()

End Sub

Private Sub FrmnewuserToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmnewuserToolStripMenuItem.Click

frmnewuser.Show()

Me.Hide()

End Sub

Private Sub FrmjetairwayseventToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmjetairwayseventToolStripMenuItem.Click

frmjetairwaysevents.Show()

Me.Hide()

End Sub

Private Sub FrmweatherToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FrmweatherToolStripMenuItem.Click

frmweather.Show()

Me.Hide()

End Sub

Private Sub FempaymentslipToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FempaymentslipToolStripMenuItem.Click

frmpaymentslip.Show()

Me.Hide()

End Sub

Private Sub DispatchbriefingreportToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DispatchbriefingreportToolStripMenuItem.Click

dispatchbriefing\_report.Show()

Me.Hide()

End Sub

Private Sub WeatherreportToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WeatherreportToolStripMenuItem.Click

weather\_report.Show()

Me.Hide()

End Sub

Private Sub PayslipreportToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PayslipreportToolStripMenuItem.Click

payslip\_report.Show()

Me.Hide()

End Sub

Private Sub FuelManagementreportToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FuelManagementreportToolStripMenuItem.Click

fuelmanagement\_report.Show()

Me.Hide()

End Sub

Private Sub frmeditattendanceToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles frmeditattendanceToolStripMenuItem.Click

frmeditattendance.Show()

Me.Hide()

End Sub

Private Sub FlightscedulereportToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FlightscedulereportToolStripMenuItem.Click

flightscedule\_report.Show()

Me.Hide()

End Sub

Private Sub frmattendanceToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles frmattendanceToolStripMenuItem.Click

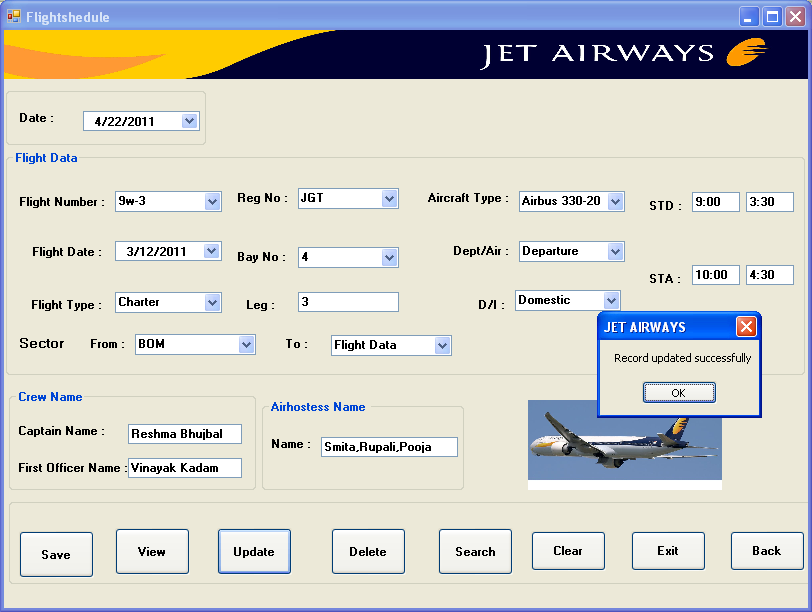
Me.Hide()

frmattendance.Show()

End Sub

End Class

**FLIGHT SCHEDULE**



**Coding:**

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmflightshedule

Private Sub frmflightshedule\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

cn.Open()

Dim cmdview As New SqlCommand("select \* from Flightschedule", cn)

Dim drview As SqlDataReader = cmdview.ExecuteReader

While drview.Read

cboflightnumber.Items.Add(drview.GetString(0))

End While

drview.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cndsave As New SqlCommand("insert into Flightschedule values('" & cboflightnumber.Text & "','" & DateTimePicker2.Value & "','" & cboregno.Text & "','" & cbobayno.Text & "','" & txtleg.Text & "','" & cboaircrafttype.Text & "','" & cboflighttype.Text & "','" & cbodi.Text & "','" & cbofrom1.Text & "','" & GroupBox2.Text & "','" & txtstd1.Text & "','" & txtstd2.Text & "','" & txtsta1.Text & "','" & txtsta2.Text & "','" & txtcaptainname.Text & "','" & txtfirstofficername.Text & "','" & txtairhostessname.Text & "','" & cbodeptarr.Text & "')", cn)

cn.Open()

cndsave.ExecuteNonQuery()

MsgBox("Record saved successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cboflightnumber\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboflightnumber.SelectedIndexChanged

Try

Dim cmd As New SqlCommand("select \* from Flightschedule where Flight\_no='" & cboflightnumber.Text & "'", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

DateTimePicker2.Text = dr.GetString(1)

cboregno.Text = dr.GetString(2)

cbobayno.Text = dr.GetString(3)

txtleg.Text = dr.GetString(4)

cboaircrafttype.Text = dr.GetString(5)

cboflighttype.Text = dr.GetString(6)

cbodi.Text = dr.GetString(7)

cbofrom1.Text = dr.GetString(8)

cboto1.Text = dr.GetString(9)

txtstd1.Text = dr.GetString(10)

txtstd2.Text = dr.GetString(11)

txtsta1.Text = dr.GetString(12)

txtsta2.Text = dr.GetString(13)

txtcaptainname.Text = dr.GetString(14)

txtfirstofficername.Text = dr.GetString(15)

txtairhostessname.Text = dr.GetString(16)

cbodeptarr.Text = dr.GetString(17)

End While

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

cboflightnumber.Text = ""

DateTimePicker1.Value = "3/12/2011"

cboregno.Items.Clear()

cbobayno.Items.Clear()

txtleg.Text = ""

cboaircrafttype.Text = ""

cboflighttype.Text = ""

cbodi.Text = ""

cbofrom1.Text = ""

cboto1.Text = ""

txtstd1.Text = ""

txtstd2.Text = ""

txtsta1.Text = ""

txtsta2.Text = ""

txtcaptainname.Text = ""

txtfirstofficername.Text = ""

txtairhostessname.Text = ""

cbodeptarr.Text = ""

End Sub

Private Sub btnupdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnupdate.Click

Try

Dim md As New SqlCommand("update Flightschedule set Flight\_date='" & DateTimePicker2.Value & "',Reg\_no='" & cboregno.Text & "',Bay\_no='" & cbobayno.Text & "',Leg\_no='" & txtleg.Text & "',Aircraft\_type='" & cboaircrafttype.Text & "', Flight\_type='" & cboflighttype.Text & "', DI='" & cbodi.Text & "',From1='" & cbofrom1.Text & "',To1='" & cboto1.Text & "',STD\_IST='" & txtstd1.Text & "',STD\_UTC='" & txtstd2.Text & "',STA\_IST='" & txtsta1.Text & "',STA\_UTC='" & txtsta2.Text & "',Captain\_name='" & txtcaptainname.Text & "',First\_off\_name='" & txtfirstofficername.Text & "',Airhostess\_name='" & txtairhostessname.Text & "',Dep\_Arr='" & cbodeptarr.Text & "' where Flight\_no='" & cboflightnumber.Text & "'", cn)

cn.Open()

md.ExecuteNonQuery()

MsgBox("Record updated successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

Try

Dim cmd As New SqlCommand("Delete from Flightschedule where Flight\_no = '" & cboflightnumber.Text & "'", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox("Record deleted successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsearch.Click

frmsearchflightentry.Show()

Me.Hide()

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub txtstd1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtstd1.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from Flightschedule", cn)

If txtstd1.Text = "" Then

txtstd2.Text = ""

Else

hour = Val(Mid(txtstd1.Text, 1, 2))

minute = Val(Mid(txtstd1.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtstd2.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtsta1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtsta1.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from Flightschedule", cn)

If txtsta1.Text = "" Then

txtsta2.Text = ""

Else

hour = Val(Mid(txtsta1.Text, 1, 2))

minute = Val(Mid(txtsta1.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtsta2.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnview.Click

Try

Dim cmd As New SqlCommand("select \* from Flightschedule where Flight\_no='" & cboflightnumber.Text & "'", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

DateTimePicker2.Text = dr.GetString(1)

cboregno.Text = dr.GetString(2)

cbobayno.Text = dr.GetString(3)

txtleg.Text = dr.GetString(4)

cboaircrafttype.Text = dr.GetString(5)

cboflighttype.Text = dr.GetString(6)

cbodi.Text = dr.GetString(7)

cbofrom1.Text = dr.GetString(8)

cboto1.Text = dr.GetString(9)

txtstd1.Text = dr.GetString(10)

txtstd2.Text = dr.GetString(11)

txtsta1.Text = dr.GetString(12)

txtsta2.Text = dr.GetString(13)

txtcaptainname.Text = dr.GetString(14)

txtfirstofficername.Text = dr.GetString(15)

txtairhostessname.Text = dr.GetString(16)

cbodeptarr.Text = dr.GetString(17)

End While

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cboregno\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles cboregno.KeyPress

Try

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtcaptainname\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtcaptainname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtfirstofficername\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtfirstofficername.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtairhostessname\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtairhostessname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtleg\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtleg.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

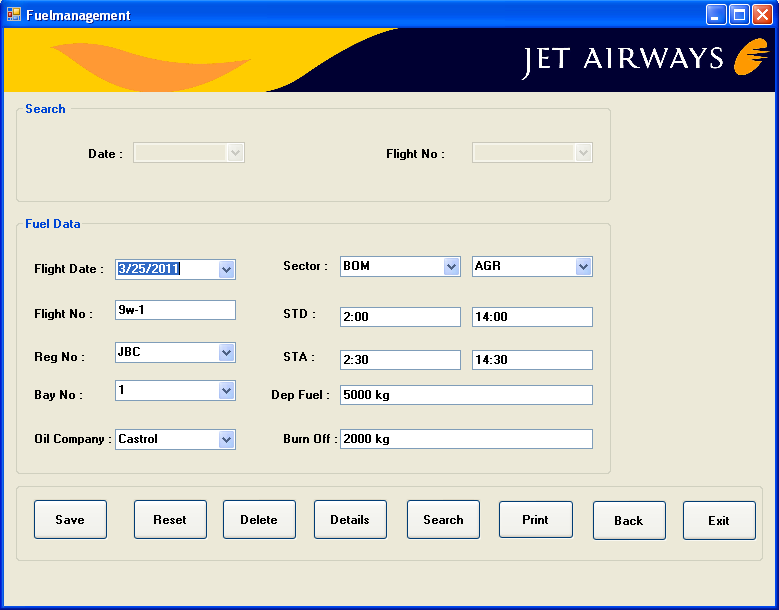
MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

End Class

**FUEL MANAGEMENT**



Coding:

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmfuelmanagement

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cmd As New SqlCommand("insert into FuelManagement values('" & txtflightno.Text & "','" & cboflightdate.Text & "','" & cboregno.Text & "','" & cbobayno.Text & "','" & cbofrom1.Text & "','" & cboto1.Text & "','" & cbooilcompany.Text & "','" & txtdeptimeIST.Text & "','" & txtdeptimeUTC.Text & "','" & txtdepfuel.Text & "','" & txtburnoff.Text & "','" & txtarrtimeIST.Text & "','" & txtarrtimeUTC.Text & "')", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox("Record saved successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "saves")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnreset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnreset.Click

txtflightno.Text = ""

cboflightdate.Text = ""

cboregno.Text = ""

cbobayno.Text = ""

cbofrom1.Text = ""

cboto1.Text = ""

cbooilcompany.Text = ""

txtdeptimeIST.Text = ""

txtdeptimeUTC.Text = ""

txtdepfuel.Text = ""

txtburnoff.Text = ""

txtarrtimeIST.Text = ""

txtarrtimeUTC.Text = ""

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

Try

Dim cmd As New SqlCommand("Delete from FuelManagement where Flight\_date = '" & cboflightdate.Text & "'", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox("Record deleted successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btndetails\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndetails.Click

Try

Dim cmd1 As New SqlCommand("select \* from FuelManagement where Flight\_date='" & cbodate.Text & "' and Flight\_no='" & cboflightno.Text & "'", cn)

cn.Open()

Dim dm As SqlDataReader = cmd1.ExecuteReader

While dm.Read

' cboflightno.Text = dm.GetString(0)

'cboflightdate.Text = dm.GetString(1)

cboregno.Text = dm.GetString(2)

cbobayno.Text = dm.GetString(3)

cbofrom1.Text = dm.GetString(4)

cboto1.Text = dm.GetString(5)

cbooilcompany.Text = dm.GetString(6)

txtdeptimeIST.Text = dm.GetString(7)

txtdeptimeUTC.Text = dm.GetString(8)

txtdepfuel.Text = dm.GetString(9)

txtburnoff.Text = dm.GetString(10)

txtarrtimeIST.Text = dm.GetString(11)

txtarrtimeUTC.Text = dm.GetString(12)

End While

dm.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtdeptimeIST\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtdeptimeIST.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from FuelMnagement", cn)

If txtdeptimeIST.Text = "" Then

txtdeptimeUTC.Text = ""

Else

hour = Val(Mid(txtdeptimeIST.Text, 1, 2))

minute = Val(Mid(txtdeptimeIST.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtdeptimeUTC.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsearch\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsearch.Click

cbodate.Enabled = True

cboflightno.Enabled = True

Dim am As New SqlCommand("select \* from FuelManagement", cn)

cn.Open()

Dim dm1 As SqlDataReader = am.ExecuteReader

While dm1.Read

cbodate.Items.Add(dm1(1))

cboflightno.Items.Add(0)

End While

dm1.Close()

cn.Close()

cboflightdate.Visible = False

txtflightno.Visible = False

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub cboflightdate\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboflightdate.SelectedIndexChanged

Try

If cn.State = ConnectionState.Open Then

cn.Close()

End If

Dim am As New SqlCommand("select \* from FuelManagement where Flight\_date='" & cboflightdate.Text & "'", cn)

cn.Open()

Dim dm As SqlDataReader = am.ExecuteReader

While dm.Read

txtflightno.Text = dm.GetString(0)

'cboflightdate.Text = dm.GetString(1)

cboregno.Text = dm.GetString(2)

cbobayno.Text = dm.GetString(3)

cbofrom1.Text = dm.GetString(4)

cboto1.Text = dm.GetString(5)

cbooilcompany.Text = dm.GetString(6)

txtdeptimeIST.Text = dm.GetString(7)

txtdeptimeUTC.Text = dm.GetString(8)

txtdepfuel.Text = dm.GetString(9)

txtburnoff.Text = dm.GetString(10)

txtarrtimeIST.Text = dm.GetString(11)

txtarrtimeUTC.Text = dm.GetString(12)

End While

dm.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub Form6\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

cbodate.Enabled = False

cboflightno.Enabled = False

Dim am As New SqlCommand("select \* from FuelManagement", cn)

cn.Open()

Dim dm As SqlDataReader = am.ExecuteReader

While dm.Read

cboflightno.Items.Add(dm(0))

cboflightdate.Items.Add(dm(1))

End While

dm.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cbodate\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbodate.SelectedIndexChanged

Try

Dim cmd As New SqlCommand("select \* from FuelManagement where Flight\_date='" & cbodate.Text & "' and Flight\_no= '" & cboflightno.Text & "'", cn)

cn.Open()

Dim dm As SqlDataReader = cmd.ExecuteReader

While dm.Read

cboflightno.Text = dm.GetString(0)

cboflightdate.Text = dm.GetString(1)

cboregno.Text = dm.GetString(2)

cbobayno.Text = dm.GetString(3)

cbofrom1.Text = dm.GetString(4)

cboto1.Text = dm.GetString(5)

cbooilcompany.Text = dm.GetString(6)

txtdeptimeIST.Text = dm.GetString(7)

txtdeptimeUTC.Text = dm.GetString(8)

txtdepfuel.Text = dm.GetString(9)

txtburnoff.Text = dm.GetString(10)

txtarrtimeIST.Text = dm.GetString(11)

txtarrtimeUTC.Text = dm.GetString(12)

End While

dm.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtarrtimeIST\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtarrtimeIST.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from FuelManagement", cn)

If txtarrtimeIST.Text = "" Then

txtarrtimeUTC.Text = ""

Else

hour = Val(Mid(txtarrtimeIST.Text, 1, 2))

minute = Val(Mid(txtarrtimeIST.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtarrtimeUTC.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub btnprint\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprint.Click

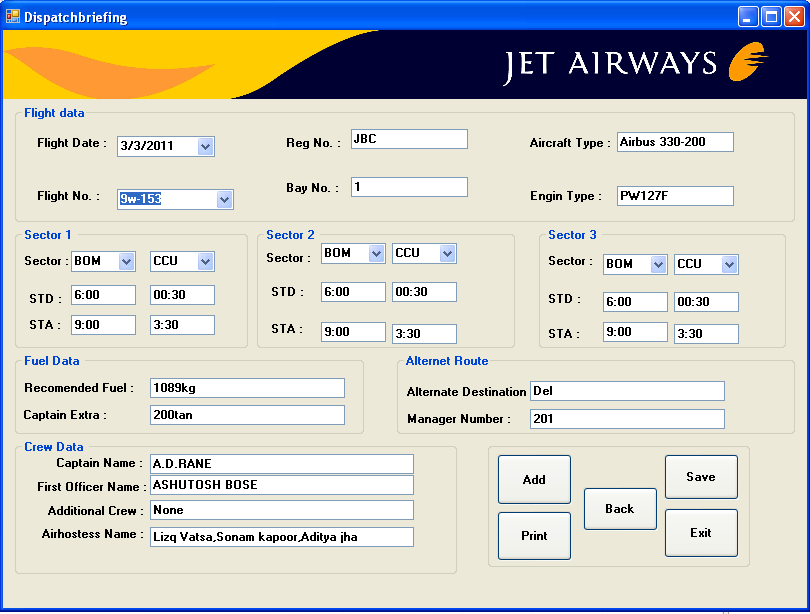
fuelmanagement\_report.Show()

Me.Hide()

End Sub

End Class

**DISPATCH BRIEFING**



Coding:

Imports System.Data.SqlClient

Imports System.Data.Sql

Public Class frmdispatchbriefing

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btnadd\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnadd.Click

cboflightno.Text = ""

cboflightdate.Text = ""

txtregno.Text = ""

txtbayno.Text = ""

txtaircrafttype.Text = ""

txtengintype.Text = ""

cbofrom1.Text = ""

cboto1.Text = ""

txtstd1.Text = ""

txtstd11.Text = ""

txtsta1.Text = ""

txtsta11.Text = ""

cbofrom2.Text = ""

cboto2.Text = ""

txtstd2.Text = ""

txtstd22.Text = ""

txtsta2.Text = ""

txtsta22.Text = ""

cbofrom3.Text = ""

cboto3.Text = ""

txtstd3.Text = ""

txtstd33.Text = ""

txtsta3.Text = ""

txtsta33.Text = ""

txtrecomendedfuel.Text = ""

txtextra.Text = ""

txtnameofcaptain.Text = ""

txtfirstofficername.Text = ""

txtadditionalcrew.Text = ""

txtairhostessname.Text = ""

txtalternatedestination.Text = ""

txtmanagerno.Text = ""

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cmd As New SqlCommand("insert into DBrief values ('" & cboflightno.Text & "','" & cboflightdate.Text & "','" & txtregno.Text & "','" & txtbayno.Text & "','" & txtaircrafttype.Text & "','" & txtengintype.Text & "','" & cbofrom1.Text & "','" & cboto1.Text & "','" & txtstd1.Text & "','" & txtstd11.Text & "','" & txtsta1.Text & "','" & txtsta11.Text & "','" & cbofrom2.Text & "','" & cboto2.Text & "','" & txtstd2.Text & "','" & txtstd22.Text & "','" & txtsta2.Text & "','" & txtsta22.Text & "','" & cbofrom3.Text & "','" & cboto3.Text & "','" & txtstd3.Text & "','" & txtstd33.Text & "','" & txtstd33.Text & "','" & txtsta3.Text & "','" & txtsta33.Text & "','" & txtrecomendedfuel.Text & "','" & txtextra.Text & "','" & txtnameofcaptain.Text & "','" & txtfirstofficername.Text & "','" & txtadditionalcrew.Text & "','" & txtairhostessname.Text & "','" & txtalternatedestination.Text & "','" & txtmanagerno.Text & "')", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox("Record saved successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "saves")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnprint\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprint.Click

flightscedule\_report.Show()

Me.Hide()

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub txtstd1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtstd1.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from DBrief", cn)

If txtstd1.Text = "" Then

txtstd11.Text = ""

Else

hour = Val(Mid(txtstd1.Text, 1, 2))

minute = Val(Mid(txtstd1.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtstd11.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtsta1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtsta1.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from DBrief", cn)

If txtsta1.Text = "" Then

txtsta11.Text = ""

Else

hour = Val(Mid(txtsta1.Text, 1, 2))

minute = Val(Mid(txtsta1.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtsta11.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtstd2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtstd2.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from DBrief", cn)

If txtstd2.Text = "" Then

txtstd22.Text = ""

Else

hour = Val(Mid(txtstd2.Text, 1, 2))

minute = Val(Mid(txtstd2.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtstd22.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtsta2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtsta2.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from DBrief", cn)

If txtsta2.Text = "" Then

txtsta22.Text = ""

Else

hour = Val(Mid(txtsta2.Text, 1, 2))

minute = Val(Mid(txtsta2.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtsta22.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtstd3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtstd3.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from DBrief", cn)

If txtstd3.Text = "" Then

txtstd33.Text = ""

Else

hour = Val(Mid(txtstd3.Text, 1, 2))

minute = Val(Mid(txtstd3.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtstd33.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtsta3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtsta3.TextChanged

Try

Dim hour, minute As Integer

Dim cmd As New SqlCommand("select \* from DBrief", cn)

If txtsta3.Text = "" Then

txtsta33.Text = ""

Else

hour = Val(Mid(txtsta3.Text, 1, 2))

minute = Val(Mid(txtsta3.Text, 2, 2))

minute = minute - 30

If (minute < 0) Then

hour = hour - 1

minute = minute + 60

End If

hour = hour - 5

If (hour < 0) Then

hour = 24 + hour

End If

txtsta33.Text = hour & ":" & minute

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cboflightno\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboflightno.SelectedIndexChanged

Try

Dim cmd As New SqlCommand("select \* from DBrief where Flight\_no='" & cboflightno.Text & "'", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cboflightno.Text = dr.GetString(0)

cboflightdate.Text = dr.GetString(1)

txtregno.Text = dr.GetString(2)

txtbayno.Text = dr.GetString(3)

txtaircrafttype.Text = dr.GetString(4)

txtengintype.Text = dr.GetString(5)

cbofrom1.Text = dr.GetString(6)

cboto1.Text = dr.GetString(7)

txtstd1.Text = dr.GetString(8)

txtstd11.Text = dr.GetString(9)

txtsta1.Text = dr.GetString(10)

txtsta11.Text = dr.GetString(11)

cbofrom2.Text = dr.GetString(12)

cboto2.Text = dr.GetString(13)

txtstd2.Text = dr.GetString(14)

txtstd22.Text = dr.GetString(15)

txtsta2.Text = dr.GetString(16)

txtsta22.Text = dr.GetString(17)

cbofrom3.Text = dr.GetString(18)

cboto3.Text = dr.GetString(19)

txtstd3.Text = dr.GetString(20)

txtstd33.Text = dr.GetString(21)

txtsta3.Text = dr.GetString(22)

txtsta33.Text = dr.GetString(23)

txtrecomendedfuel.Text = dr.GetString(24)

txtextra.Text = dr.GetString(25)

txtnameofcaptain.Text = dr.GetString(26)

txtfirstofficername.Text = dr.GetString(27)

txtadditionalcrew.Text = dr.GetString(28)

txtairhostessname.Text = dr.GetString(29)

txtalternatedestination.Text = dr.GetString(30)

txtmanagerno.Text = dr.GetString(32)

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub frmdispatchbriefing\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

Dim cmd As New SqlCommand("select \* from DBrief", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cboflightno.Items.Add(dr.GetString(0))

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtregno\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtregno.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtnameofcaptain\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtnameofcaptain.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtfirstofficername\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtfirstofficername.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtadditionalcrew\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtadditionalcrew.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtairhostessname\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtairhostessname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtalternatedestination\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtalternatedestination.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

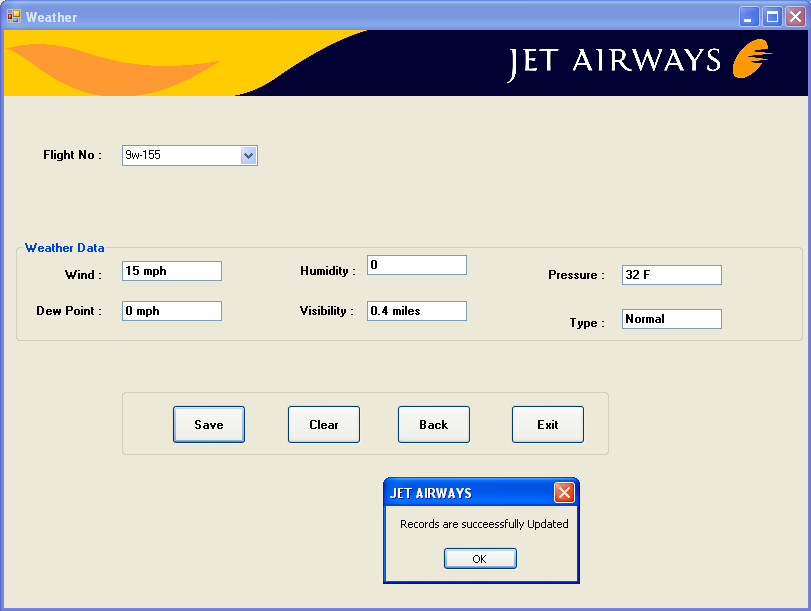
MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

End Class

**WEATHER**



Coding:

Imports System.Data

Imports System.Data.SqlClient

Public Class frmweather

Private Sub frmweather\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

Dim cmd As New SqlCommand("select \* from DBrief", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cboflightno.Items.Add(dr.GetString(0))

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

Try

txtwind.Text = ""

txtdewpoint.Text = ""

txthumidity.Text = ""

txtvisibility.Text = ""

txtpressure.Text = ""

txttype.Text = ""

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim sd As New SqlCommand("update DBrief set Wind='" & txtwind.Text & "' ,Humidity='" & txthumidity.Text & "' , Pressure='" & txtpressure.Text & "' ,Dew ='" & txtdewpoint.Text & "' , Visibility='" & txtvisibility.Text & "' , Type='" & txttype.Text & "' where Flight\_no= '" & cboflightno.Text & "'", cn)

cn.Open()

sd.ExecuteNonQuery()

MsgBox("Records are succeessfully Updated")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

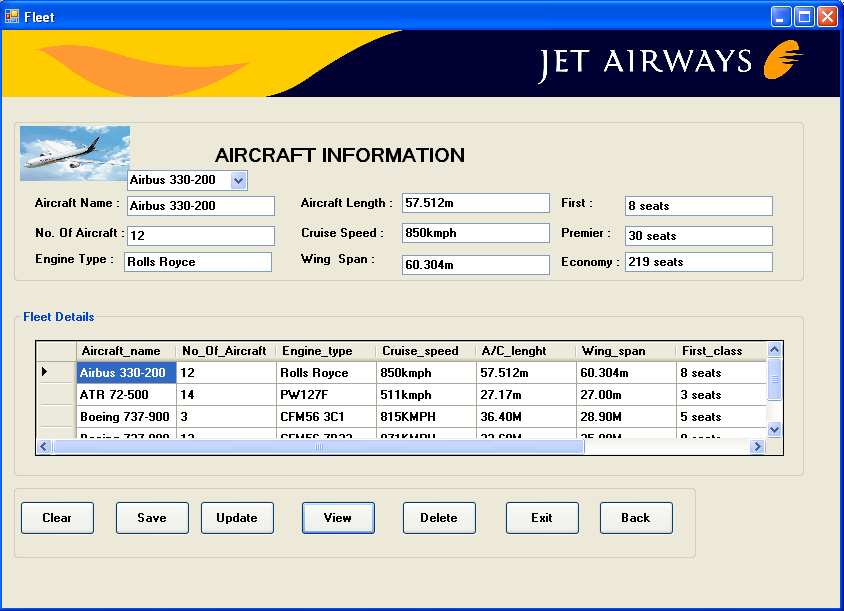
End If

End Try

End Sub

End Class

**FLEET**



Coding :

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmaircraftdata

Private Sub btnview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnview.Click

Try

Dim da As New SqlDataAdapter("select \* from Fleet", cn)

Dim ds As New DataSet

da.Fill(ds, "Fleet")

DataGridView1.DataSource = ds.Tables(0)

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cndsave As New SqlCommand("insert into Fleet values('" & txtaircraftname.Text & "','" & txtnoofaircraft.Text & "','" & txteginetype.Text & "','" & txtaircraftlength.Text & "','" & txtcruisespeed.Text & "','" & txtwingspan.Text & "','" & txtfirst.Text & "','" & txtpremier.Text & "','" & txteconomy.Text & "')", cn)

cn.Open()

cndsave.ExecuteNonQuery()

MsgBox("Records saved successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

Try

Dim cmd As New SqlCommand("Delete from Fleet where Aircraft\_name='" & txtaircraftname.Text & "'", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox("Record deleted successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub frmfleet\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

'cn.Open()

'Dim da As New SqlDataAdapter("select \* from Fleet", cn)

'Dim ds As New DataSet

'da.Fill(ds, "Fleet")

'DataGridView1.DataSource = ds.Tables(0)

Try

Dim cmd As New SqlCommand("select \* from Fleet", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cboaircraftname.Items.Add(dr(0))

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnupdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnupdate.Click

Try

Dim sd As New SqlCommand("update Fleet set No\_Of\_Aircraft='" & txtnoofaircraft.Text & "' Engine\_type='" & txteginetype.Text & "' Cruise\_speed='" & txtcruisespeed.Text & "' & A/C length='" & txtaircraftlength.Text & "' & Wing\_span='" & txtwingspan.Text & "' & First\_class='" & txtfirst.Text & "' & Premier\_class='" & txtpremier.Text & "' & Economy\_class='" & txteconomy.Text & "' where Aircraft\_name= '" & txtaircraftname.Text & "' )", cn)

cn.Open()

MsgBox("Records are succeessfully Updated")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cboaircraftname\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboaircraftname.SelectedIndexChanged

Try

Dim cmd As New SqlCommand("select \* from Fleet where Aircraft\_name='" & cboaircraftname.Text & "'", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

txtaircraftname.Text = dr.GetString(0)

cboaircraftname.Text = dr.GetString(0)

txtnoofaircraft.Text = dr.GetString(1)

txteginetype.Text = dr.GetString(2)

txtcruisespeed.Text = dr.GetString(3)

txtaircraftlength.Text = dr.GetString(4)

txtwingspan.Text = dr.GetString(5)

txtfirst.Text = dr.GetString(6)

txtpremier.Text = dr.GetString(7)

txteconomy.Text = dr.GetString(8)

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

txtaircraftname.Text = ""

cboaircraftname.Text = ""

txtaircraftlength.Text = ""

txtcruisespeed.Text = ""

txteconomy.Text = ""

txteginetype.Text = ""

txtfirst.Text = ""

txtnoofaircraft.Text = ""

txtpremier.Text = ""

txtwingspan.Text = ""

End Sub

Private Sub txtnoofaircraft\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtnoofaircraft.KeyPress

Try

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

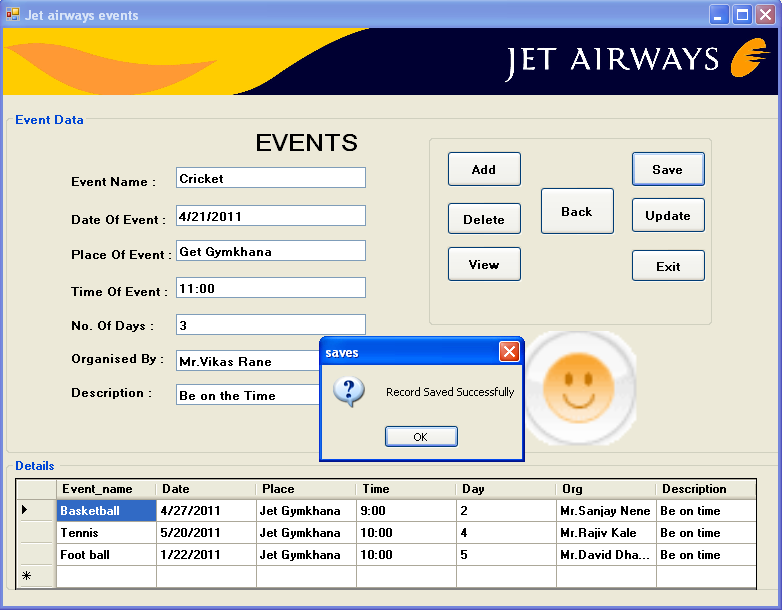
MDIParent1.Show()

Me.Hide()

End Sub

End Class

**JET AIRWAYS EVENTS**



Coding :

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmjetairwaysevents

Private Sub btnadd\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnadd.Click

txteventname.Text = ""

txteventdate.Text = ""

txtplace.Text = ""

txttimeofevent.Text = ""

txtnoofevent.Text = ""

txtorganisedby.Text = ""

txtdescrition.Text = ""

End Sub

Private Sub btnview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnview.Click

Try

Dim da As New SqlDataAdapter("select \* from Event", cn)

Dim ds As New DataSet

da.Fill(ds, "Event")

DataGridView1.DataSource = ds.Tables("Event")

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cmd As New SqlCommand("insert into Event values ('" & txteventname.Text & "','" & txteventdate.Text & "','" & txtplace.Text & "','" & txttimeofevent.Text & "','" & txtnoofevent.Text & "','" & txtorganisedby.Text & "','" & txtdescrition.Text & "')", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox(" Record Saved Successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "saves")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btnupdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnupdate.Click

Try

Dim cnd As New SqlCommand("update Event set Event\_name='" & txteventname.Text & "',Date='" & txteventdate.Text & "',Place='" & txtplace.Text & "' , Time='" & txttimeofevent.Text & "',Day='" & txtnoofevent.Text & "', Org='" & txtorganisedby.Text & "', Description='" & txtdescrition.Text & "' where Event\_name='" & txteventname.Text & "' )", cn)

cn.Open()

cnd.ExecuteNonQuery()

MsgBox(" Record Updated Successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "Update")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

Try

Dim cmd As New SqlCommand("delete from Event where Event\_name='" & txteventname.Text & "'", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox(" Record Deleted Successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "Delete")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub txteventname\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txteventname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtnoofevent\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtnoofevent.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

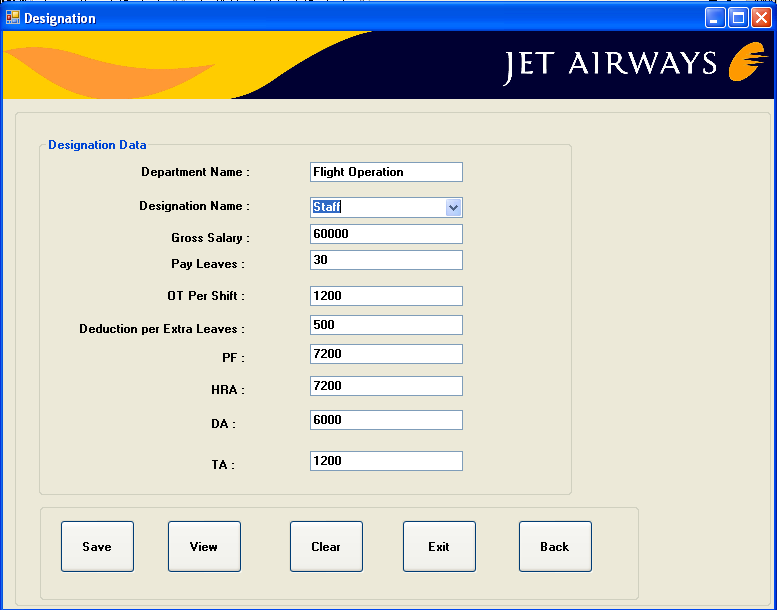
MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

End Class

**DESIGNATION**



Coding :

Imports System.Data

Imports System.Data.SqlClient

Public Class frmdesignation

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

txtdepartmentname.Text = ""

cbodesignationname.Text = ""

txtgrosssalary.Text = ""

txtpayleaves.Text = ""

txtotpershift.Text = ""

txtdeductionperextraleaves.Text = ""

txtPF.Text = ""

txtHRA.Text = ""

txtDA.Text = ""

txtTA.Text = ""

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btnview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnview.Click

Try

Dim cmd As New SqlCommand("select \* from Designation", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cbodesignationname.Items.Add(dr(0))

End While

'txtdeductionperextraleaves.Text = ""

'txtdepartmentname.Text = ""

'txtgrosssalary.Text = ""

txtdepartmentname.Text = ""

cbodesignationname.Text = ""

txtgrosssalary.Text = ""

txtpayleaves.Text = ""

txtotpershift.Text = ""

txtdeductionperextraleaves.Text = ""

txtPF.Text = ""

txtHRA.Text = ""

txtDA.Text = ""

txtTA.Text = ""

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cbodesignationname\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbodesignationname.SelectedIndexChanged

Try

Dim cmd As New SqlCommand("select \* from Designation where Designation ='" & cbodesignationname.Text & "'", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

txtdepartmentname.Text = dr(0)

txtgrosssalary.Text = dr(2)

txtpayleaves.Text = dr(3)

txtotpershift.Text = dr(4)

txtdeductionperextraleaves.Text = dr(5)

txtPF.Text = dr(6)

txtHRA.Text = dr(7)

txtDA.Text = dr(8)

txtTA.Text = dr(9)

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub frmdesignation\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

Dim cmd As New SqlCommand("select \* from Designation", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cbodesignationname.Items.Add(dr(1))

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cmd As New SqlCommand("insert into Designation values ('" & txtdepartmentname.Text & "','" & cbodesignationname.Text & "','" & txtgrosssalary.Text & "','" & txtpayleaves.Text & "','" & txtotpershift.Text & "','" & txtdeductionperextraleaves.Text & "','" & txtPF.Text & "','" & txtHRA.Text & "','" & txtDA.Text & "','" & txtTA.Text & "')", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox(" Record Saved Successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "saves")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub txtdepartmentname\_keypress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtdepartmentname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtgrosssalary\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtgrosssalary.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtpayleaves\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtpayleaves.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtotpershift\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtotpershift.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtdeductionperextraleaves\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtdeductionperextraleaves.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtPF\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPF.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtHRA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtHRA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtDA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtDA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtTA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtTA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

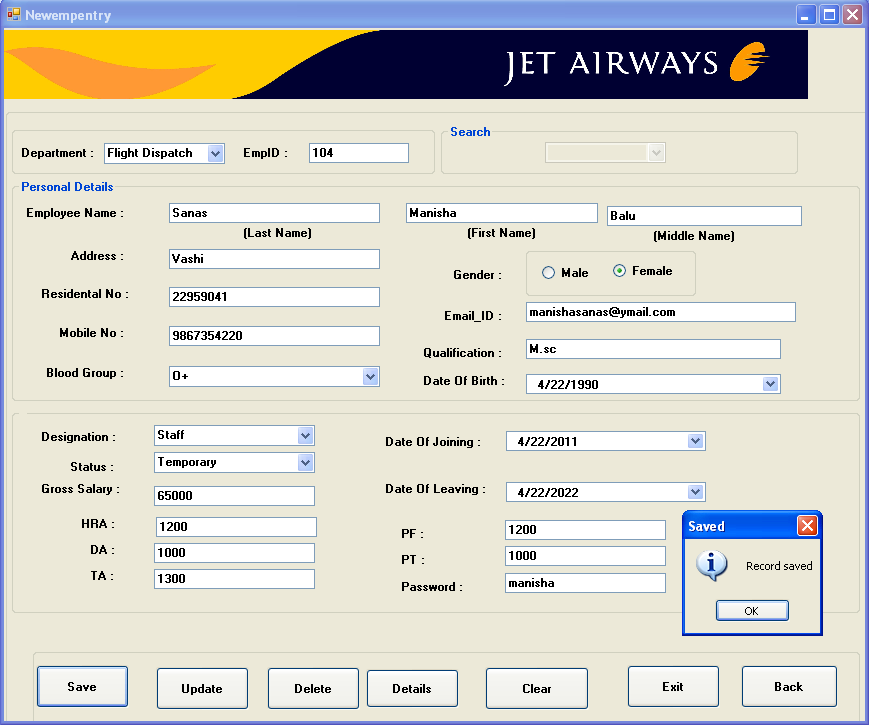
MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

End Class

**NEW EMPLOYEE ENTRY**



Coding:

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmnewemployeeentry

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cndsave As New SqlCommand("insert into Employee\_entry values (" & txtempid.Text & ",'" & txtlastname.Text & "','" & txtfirstname.Text & "','" & txtmiddlename.Text & "','" & txtaddress.Text & "','" & txtresidentalno.Text & "','" & txtmobileno.Text & "','" & txtgender.Text & "','" & txtemailid.Text & "','" & cbobloodgroup.Text & "','" & txtqualification.Text & "','" & dtpbirth\_date.Value.Date & "','" & dtpdat\_of\_joining.Value.Date & "','" & dtpdate\_of\_leaving.Value.Date & "','" & cbodesignation.Text & "','" & cbodepartment.Text & "','" & cbostatus.Text & "','" & txtgrosssalary.Text & "','" & txtHRA.Text & "','" & txtDA.Text & "','" & txtTA.Text & "','" & txtPF.Text & "','" & txtPT.Text & "','" & txtpassword.Text & "')", cn)

cn.Open()

cndsave.ExecuteNonQuery()

MsgBox("Record saved", MsgBoxStyle.Information, "Saved")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub frmnewempentry\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

Dim cmdauto As New SqlCommand("select \* from Employee\_entry ", cn)

cn.Open()

Dim dr As SqlDataReader = cmdauto.ExecuteReader

While dr.Read

txtempid.Text = dr.GetString(0)

End While

txtempid.Text = Val(txtempid.Text + 1)

cboempid.Enabled = False

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub rbtnfemale\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles rbtnfemale.Click

txtgender.Text = "Female"

End Sub

Private Sub rbtnmale\_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles rbtnmale.Click

txtgender.Text = "Male"

End Sub

Private Sub btnupdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnupdate.Click

Try

Dim sd As New SqlCommand("update Employee\_entry set Employee\_last\_name='" & txtlastname.Text & "' ,Employee\_first\_name='" & txtfirstname.Text & "' , Employee\_middle\_name='" & txtmiddlename.Text & "' ,Address ='" & txtaddress.Text & "' , Residential\_no='" & txtresidentalno.Text & "' , Mobile\_no='" & txtmobileno.Text & "' , Gender='" & txtgender.Text & "', Email\_id='" & txtemailid.Text & "' , Blood\_group= '" & cbobloodgroup.Text & "' , Qualification='" & txtqualification.Text & "' , DOB='" & dtpbirth\_date.Text & "', DOJ='" & dtpdat\_of\_joining.Text & "' ,DOL='" & dtpdate\_of\_leaving.Text & "', Designation='" & cbodesignation.Text & "',Department='" & cbodepartment.Text & "' , Status='" & cbostatus.Text & "' , Gross\_salary='" & txtgrosssalary.Text & "' HRA='" & txtHRA.Text & "',DA='" & txtDA.Text & "',TA='" & txtTA.Text & "',PF='" & txtPF.Text & "',PT='" & txtPT.Text & "',Password='" & txtpassword.Text & "' where Employee\_id= '" & cboempid.Text & "'", cn)

cn.Open()

sd.ExecuteNonQuery()

MsgBox("Records are succeessfully Updated")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btndetails\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndetails.Click

cboempid.Enabled = True

Dim cmdfill As New SqlCommand("select \* from Employee\_entry", cn)

cn.Open()

Dim drfill As SqlDataReader = cmdfill.ExecuteReader

While drfill.Read

cboempid.Items.Add(drfill.GetString(0))

End While

drfill.Close()

cn.Close()

txtempid.Enabled = False

cboempid.Visible = True

MsgBox("Select approriate Employee\_ID from combobox which you want", MsgBoxStyle.OkOnly + MsgBoxStyle.Question)

cboempid.Enabled = True

cboempid.Focus()

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub cboempid\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboempid.SelectedIndexChanged

Try

Dim cmd As New SqlCommand("select \* from Employee\_entry where Employee\_id='" & cboempid.Text & "'", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

txtempid.Text = dr.GetString(0)

cboempid.Text = dr.GetString(0)

txtlastname.Text = dr.GetString(1)

txtfirstname.Text = dr.GetString(2)

txtmiddlename.Text = dr.GetString(3)

txtaddress.Text = dr.GetString(4)

txtresidentalno.Text = dr.GetString(5)

txtmobileno.Text = dr.GetString(6)

txtgender.Text = dr.GetString(7)

txtemailid.Text = dr.GetString(8)

cbobloodgroup.Text = dr.GetString(9)

txtqualification.Text = dr.GetString(10)

dtpbirth\_date.Value = dr(11)

dtpdat\_of\_joining.Value = dr(12)

dtpdate\_of\_leaving.Value = dr(13)

cbodesignation.Text = dr.GetString(14)

cbodepartment.Text = dr.GetString(15)

cbostatus.Text = dr.GetString(16)

txtgrosssalary.Text = dr.GetString(17)

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

Try

cboempid.Visible = True

Dim cmd As New SqlCommand("Delete from Employee\_entry where Employee\_id='" & cboempid.Text & "'", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox("Record deleted successfully")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

txtempid.Text = ""

cboempid.Text = ""

txtlastname.Text = ""

txtfirstname.Text = ""

txtmiddlename.Text = ""

txtaddress.Text = ""

txtresidentalno.Text = ""

txtmobileno.Text = ""

txtgender.Text = ""

txtemailid.Text = ""

cbobloodgroup.Text = ""

txtqualification.Text = ""

dtpbirth\_date.Value = Now

dtpdat\_of\_joining.Value = Now

dtpdate\_of\_leaving.Value = Now

cbodesignation.Text = ""

cbodepartment.Text = ""

cbostatus.Text = ""

txtgrosssalary.Text = ""

End Sub

Private Sub txtresidentalno\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtresidentalno.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtgrosssalary\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtgrosssalary.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtlastname\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtlastname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtfirstname\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtfirstname.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtmiddlename\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtmiddlename.KeyPress

If funValidate\_Text(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Text....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtHRA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtHRA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtTA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtTA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtPF\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPF.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtPT\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPT.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtDA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtDA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

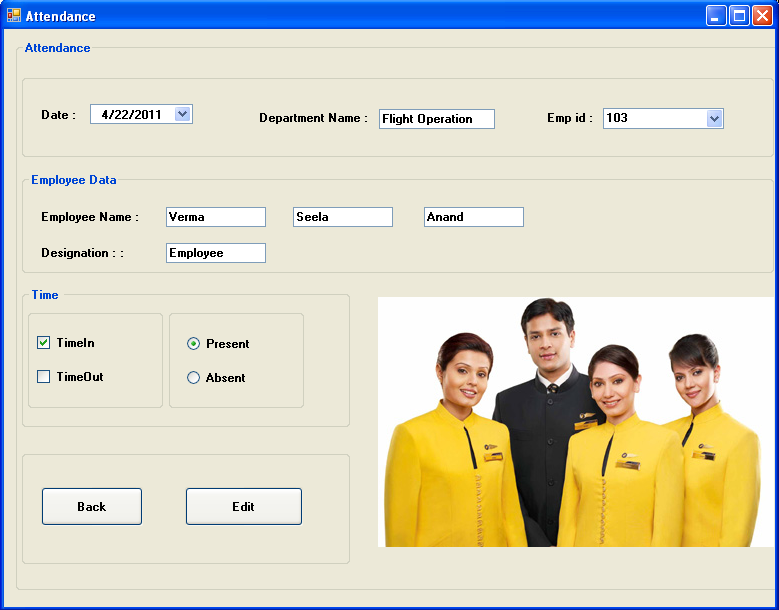
MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

End Clas

**ATTENDANCE**



Coding:

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmattendance

Private Sub frmattendance\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

DateTimePicker1.Value = Now.Date

Dim cmdemp As New SqlCommand("select \* from Employee\_entry", cn)

cn.Open()

Dim dremp As SqlDataReader = cmdemp.ExecuteReader

While dremp.Read

cboempid.Items.Add(dremp.GetString(0))

End While

dremp.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub cboempid\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboempid.SelectedIndexChanged

Try

Dim cmddept As New SqlCommand("select \* from Employee\_entry where Employee\_id='" & cboempid.Text & "'", cn)

cn.Open()

Dim drdept As SqlDataReader = cmddept.ExecuteReader

While drdept.Read

txtdeptname.Text = drdept.GetString(15)

txtlastname.Text = drdept.GetString(1)

txtmiddlename.Text = drdept.GetString(3)

txtfirstname.Text = drdept.GetString(2)

txtdesignation.Text = drdept.GetString(14)

End While

drdept.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnedit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnedit.Click

Try

If chkTimeOut.Checked = True And rbtnpresent.Checked = True Then

Dim cmdedit As New SqlCommand("select \* from Attendance where EmpId='" & cboempid.Text & "'", cn)

cn.Open()

Dim dredit As SqlDataReader = cmdedit.ExecuteReader

If Not dredit.Read Then

MsgBox("First select Time in.....", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

Else

frmeditattendance.Show()

End If

dredit.Close()

cn.Close()

Else

frmeditattendance.Show()

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

End Class

**EDIT ATTENDANCE**



Coding:

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmeditattendance

Private Sub frmeditattendance\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

txttype.Text = "PRESENT"

DateTimePicker1.Value = frmattendance.DateTimePicker1.Value

txtempid.Text = frmattendance.cboempid.Text

txttimein.Text = TimeOfDay

txttimeout.Text = DatePart(DateInterval.Hour, TimeOfDay) + 8 & ":" & DatePart(DateInterval.Minute, TimeOfDay) & ":" & DatePart(DateInterval.Second, TimeOfDay)

If frmattendance.chkTimeIn.Checked = True And frmattendance.rbtnpresent.Checked = True Then

txtot.Text = "N/A"

Else

If cn.State = ConnectionState.Open Then

cn.Close()

End If

Dim cmdedit As New SqlCommand("select \* from Attendance where EmpId='" & txtempid.Text & "'", cn)

cn.Open()

Dim dredit As SqlDataReader = cmdedit.ExecuteReader

While dredit.Read

txttimein.Text = dredit(2)

txttimeout.Text = dredit(3)

DateTimePicker2.Value = dredit.GetDateTime(3)

'TextBox1.Text = dredit(3)

End While

dredit.Close()

cn.Close()

txtot.Text = DateDiff(DateInterval.Hour, TimeOfDay, DateTimePicker2.Value)

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnok\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click

Try

If frmattendance.chkTimeIn.Checked = True And frmattendance.rbtnpresent.Checked = True Then

'txtot.Text = "N/A"

Dim cmdpresent As New SqlCommand("insert into Attendance values('" & txtempid.Text & "','" & DateTimePicker1.Value & "','" & txttimein.Text & "','" & txttimeout.Text & "','" & txtot.Text & "','" & txttype.Text & "')", cn)

cn.Open()

cmdpresent.ExecuteNonQuery()

MsgBox("Attendance marked successfully...!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "ATTENDANCE")

cn.Close()

Else

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

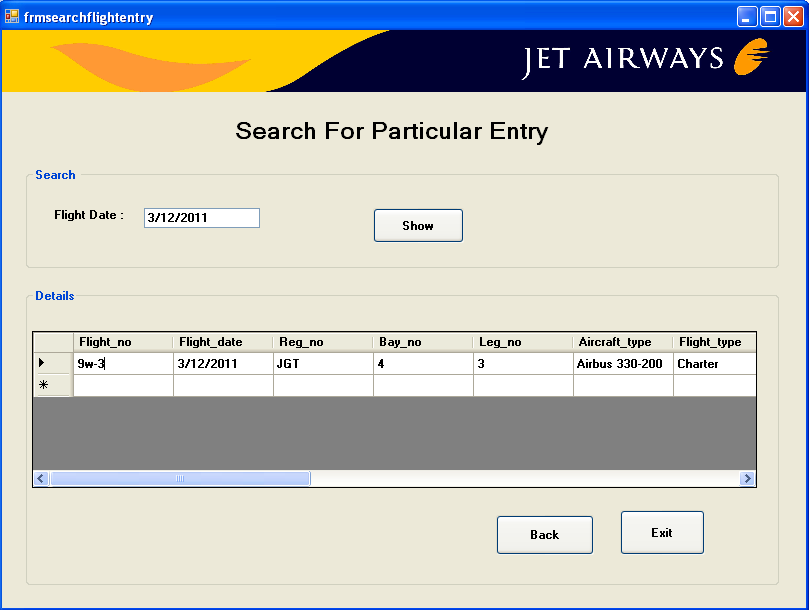
MDIParent1.Show()

Me.Hide()

End Sub

End Clas

**SEARCHE FOR PARTICULAR ENTRY**



Coding:

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmsearchflightentry

Private Sub btnshow\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnshow.Click

Try

Dim am As New SqlDataAdapter("select \* from Flightschedule where Flight\_date='" & txtdate.Text & "'", cn)

Dim dr As New DataSet

am.Fill(dr, "Flightschedule")

DataGridView1.DataSource = dr.Tables(0)

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

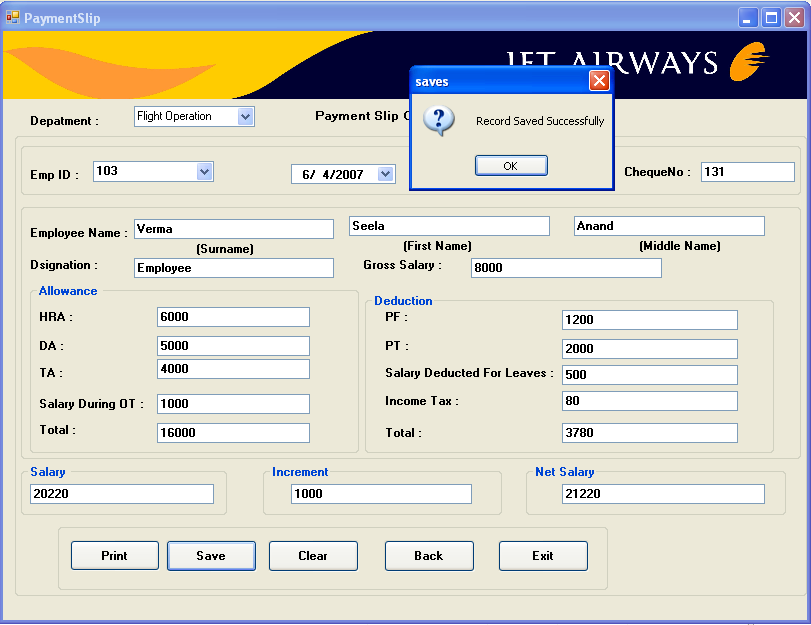
MDIParent1.Show()

Me.Hide()

End Sub

End Class

**PAYMENT SLIP**



Coding:

Imports System.Data.Sql

Imports System.Data.SqlClient

Public Class frmpaymentslip

Private Sub cboempid\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles cboempid.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub cboempid\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cboempid.SelectedIndexChanged

Try

cn.Open()

Dim cnd As New SqlCommand("select \* from Employee\_entry where Employee\_id ='" & cboempid.Text & "'", cn)

Dim dr As SqlDataReader = cnd.ExecuteReader

While dr.Read

cbodepartment.Text = dr.GetString(15)

cboempid.Text = dr.GetString(0)

txtlastname.Text = dr.GetString(1)

txtfirstname.Text = dr.GetString(2)

txtmiddlename.Text = dr.GetString(3)

txtdesignation.Text = dr.GetString(14)

txtgrosssalary.Text = dr.GetString(17)

txtHRA.Text = dr.GetString(18)

txtDA.Text = dr.GetString(19)

txtTA.Text = dr.GetString(20)

txtPF.Text = dr.GetString(21)

txtPT.Text = dr.GetString(22)

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub Form14\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

Dim cmd As New SqlCommand("select \* from Employee\_entry", cn)

cn.Open()

Dim dr As SqlDataReader = cmd.ExecuteReader

While dr.Read

cboempid.Items.Add(dr(0))

End While

dr.Close()

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtallowancetotal\_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtallowancetotal.GotFocus

txtallowancetotal.Text = Val(txtHRA.Text) + Val(txtDA.Text) + Val(txtTA.Text) + Val(txtsalaryduringOT.Text)

End Sub

Private Sub txtsalarydeductedforleaves\_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtsalarydeductedforleaves.GotFocus

Try

Dim x As String = "PRESENT"

Dim cmd1 As New SqlCommand("select count(Type) from Attendance where EmpId='" & cboempid.Text & "'and currentdate='" & DateTimePicker1.Value & "'and Type='" & x & "' and CurrentDate='" & FormatDateTime(DateTimePicker1.Value, DateFormat.ShortDate) & "'", cn)

cn.Open()

Dim counta As Integer = cmd1.ExecuteScalar

'MsgBox(counta)

txtsalarydeductedforleaves.Text = counta

txtsalarydeductedforleaves.Text = Val(txtsalarydeductedforleaves.Text) \* ((Val(txtgrosssalary.Text) / 30))

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

MDIParent1.Show()

Me.Hide()

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

End

End Sub

Private Sub txtincometax\_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtincometax.GotFocus

Try

Dim annualsalary As Double = 0.0

annualsalary = Val(txtgrosssalary.Text) \* 12

If (annualsalary >= 200000) Then

txtincometax.Text = Val(txtgrosssalary.Text) \* Val(10 / 100)

txtincometax.Text = Val(txtincometax.Text) / 12

Else

If (annualsalary > 300000 & annualsalary < 500000) Then

txtincometax.Text = Val(txtgrosssalary.Text) \* Val(12 / 100)

txtincometax.Text = Val(txtincometax.Text) / 12

Else

txtincometax.Text = Val(txtgrosssalary.Text) \* Val(15 / 100)

txtincometax.Text = Val(txtincometax.Text) / 12

End If

End If

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub txtdeductiontotal\_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtdeductiontotal.GotFocus

txtdeductiontotal.Text = Val(txtPF.Text) + Val(txtPT.Text) + Val(txtsalarydeductedforleaves.Text) + Val(txtincometax.Text)

End Sub

Private Sub txtsalary\_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtsalary.GotFocus

txtsalary.Text = (Val(txtgrosssalary.Text) + Val(txtallowancetotal.Text)) - Val(txtdeductiontotal.Text)

End Sub

Private Sub txtnetsalary\_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtnetsalary.GotFocus

txtnetsalary.Text = Val(txtsalary.Text) + Val(txtincrement.Text)

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

Try

Dim cmd As New SqlCommand("insert into PaySlip values('" & cboempid.Text & "','" & DateTimePicker1.Value & "','" & txtchequeno.Text & "','" & cbodepartment.Text & "','" & txtHRA.Text & "','" & txtDA.Text & "','" & txtTA.Text & "','" & txtsalaryduringOT.Text & "','" & txtallowancetotal.Text & "','" & txtPF.Text & "','" & txtincrement.Text & "','" & txtsalarydeductedforleaves.Text & "','" & txtincometax.Text & "','" & txtdeductiontotal.Text & "','" & txtnetsalary.Text & "','" & txtPT.Text & "')", cn)

cn.Open()

cmd.ExecuteNonQuery()

MsgBox(" Record Saved Successfully", MsgBoxStyle.OkOnly + MsgBoxStyle.Question, "saves")

cn.Close()

Catch ex As Exception

MsgBox(ex.Message, MsgBoxStyle.Critical, Me.Text)

If cn.State = ConnectionState.Open Then

cn.Close()

End If

End Try

End Sub

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

cbodepartment.Text = ""

cboempid.Text = ""

txtchequeno.Text = ""

txtlastname.Text = ""

txtfirstname.Text = ""

txtmiddlename.Text = ""

txtdesignation.Text = ""

txtgrosssalary.Text = ""

txtHRA.Text = ""

txtDA.Text = ""

txtTA.Text = ""

txtsalaryduringOT.Text = ""

txtallowancetotal.Text = ""

txtPF.Text = ""

txtPT.Text = ""

txtsalarydeductedforleaves.Text = ""

txtincometax.Text = ""

txtdeductiontotal.Text = ""

txtsalary.Text = ""

txtincrement.Text = ""

txtnetsalary.Text = ""

End Sub

Private Sub btnprint\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprint.Click

payslip\_report.Show()

Me.Hide()

End Sub

Private Sub txtgrosssalary\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtgrosssalary.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtHRA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtHRA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtDA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtDA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtTA\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtTA.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtsalaryduringOT\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtsalaryduringOT.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

End Sub

Private Sub txtincrement\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtincrement.KeyPress

If funValidate\_Number(Asc(e.KeyChar)) = False Then

e.Handled = True

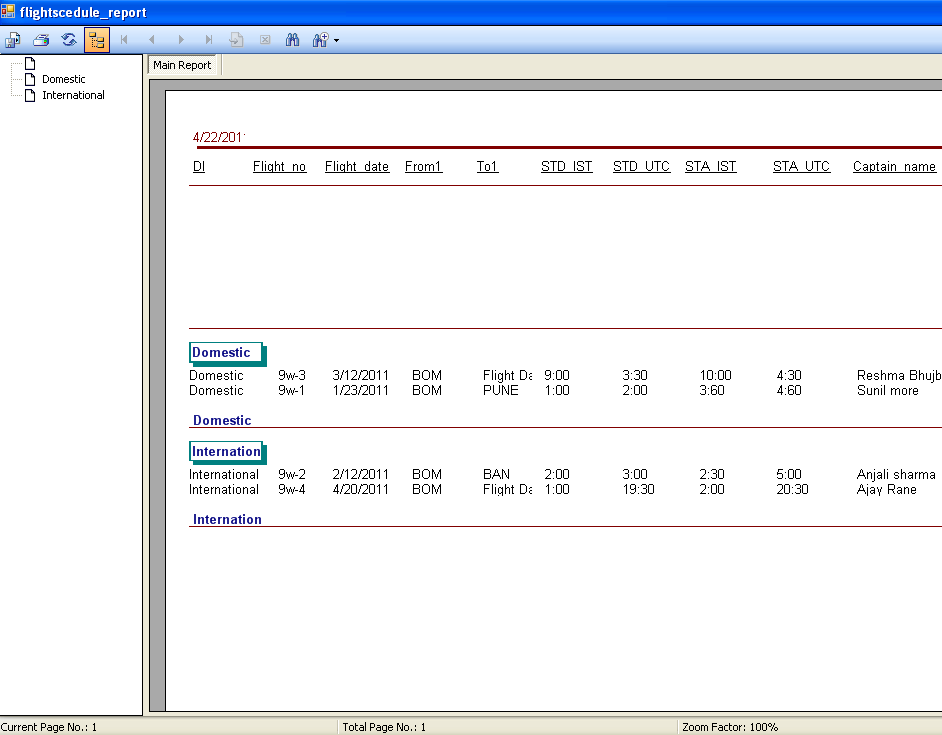
MsgBox("Please Enter Only Numbers....!", MsgBoxStyle.OkOnly + MsgBoxStyle.Critical, "ERROR")

End If

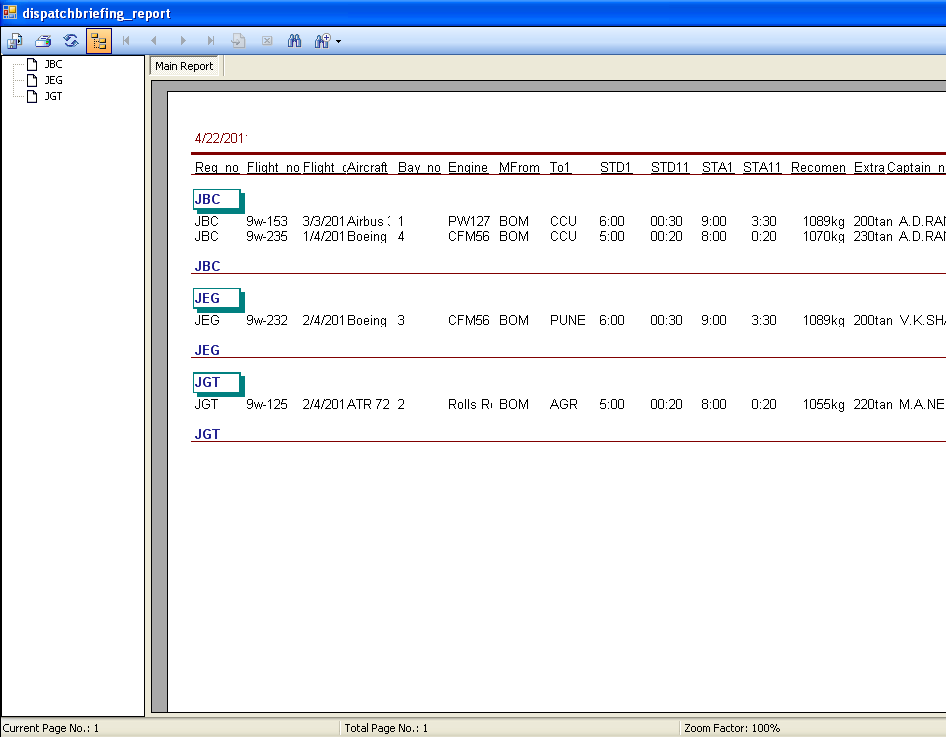
End Sub

End Class

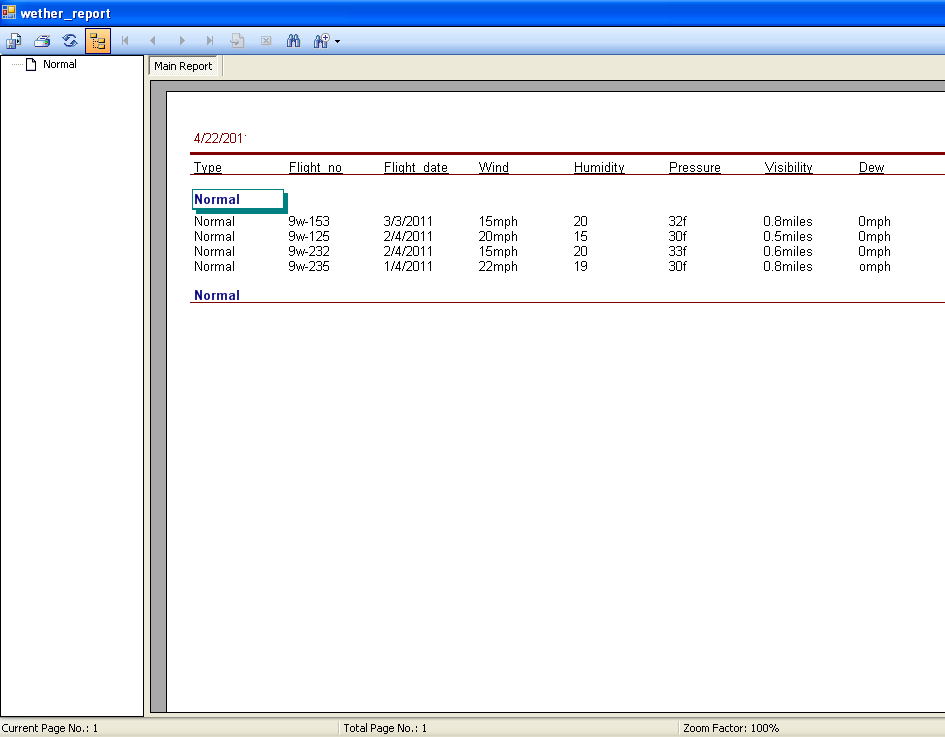
**FLIGHT SCHEDULE REPORT**



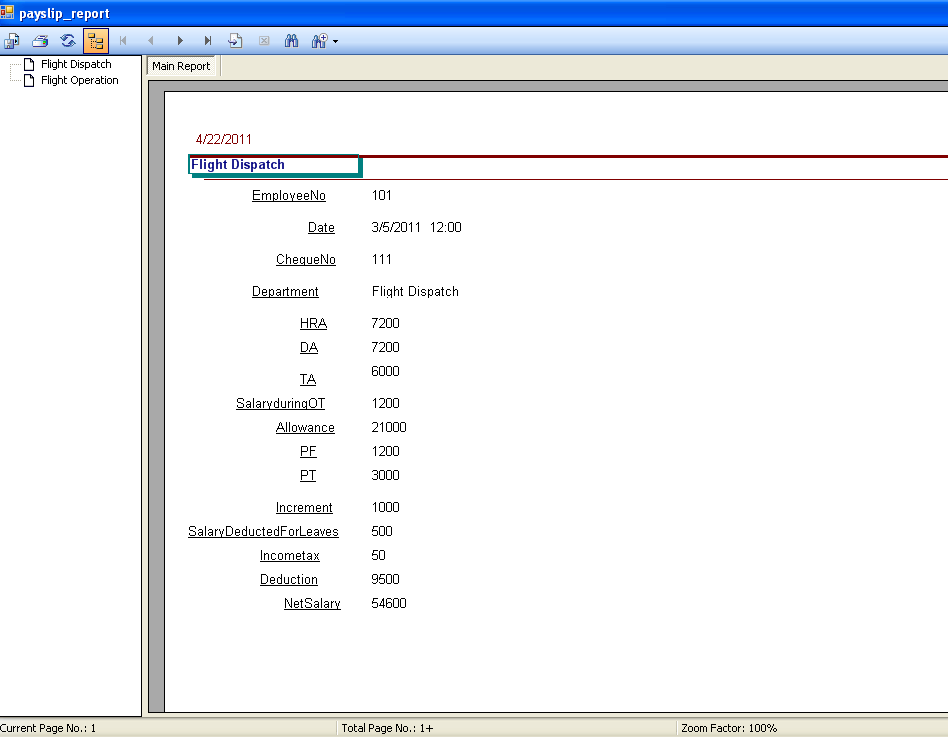
**DISPATCH BRIEFING REPORT**



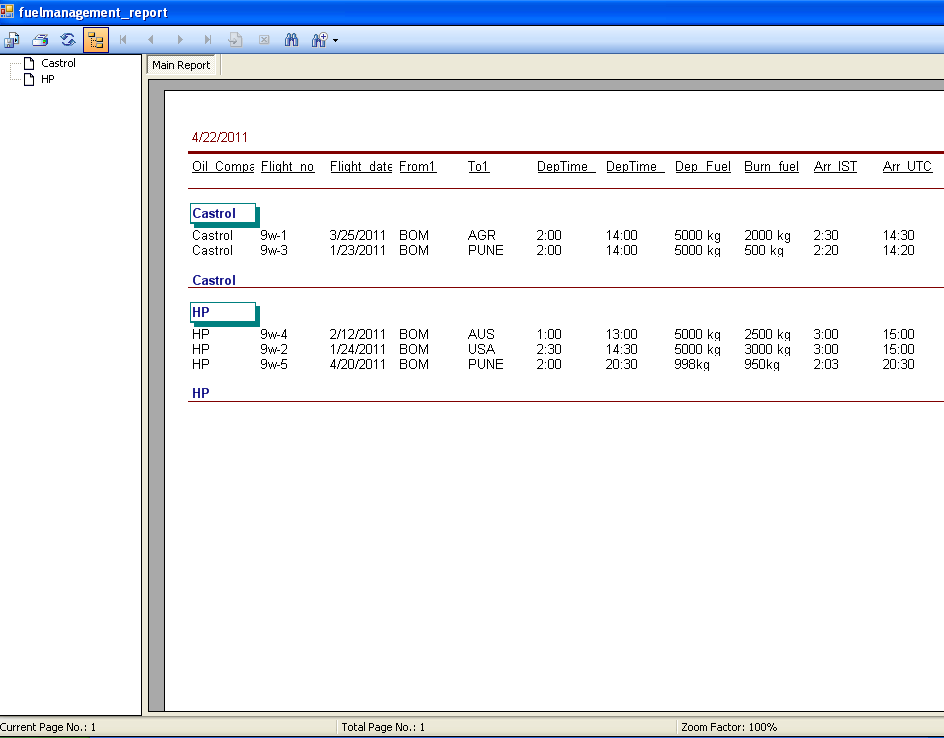
**WEATHER REPORT**



**PAYMENT SLIP REPORT**



**FUEL MANAGEMENT REPORT**



**CONCLUSION**

The project comprehensively uses vb.net to handle the customer more flexibly and to make more profit.

SQL(Structured query language) for the database connectivity. Thus I have developed this project using vb.net as Front End and MS SQL as Back End.

**BIBLIOGRAPHY**

**BOOKS REFERRED :**

* [**Visual Basic .NET How to Program (2nd Edition)**](http://visualbasic.about.com/library/bldeitel1-1a.htm) **:**  
  By Harvey M. Deitel, Paul J. Deitel, Tem R. Nieto
* [**Visual Basic .NET For Experienced Programmers**](http://visualbasic.about.com/library/bldeitel2-1a.htm) **:**By Harvey M. Deitel, Paul J. Deitel
* [**Programming Visual Basic .NET**](http://visualbasic.about.com/library/blgrundgeiger1-1a.htm) **:**By Dave Grundgeiger
* [**Learning VB.NET**](http://visualbasic.about.com/library/blliberty1-1a.htm) **:**  
  By Jesse Liberty
* **The Complete reference SQL :**

By James R.Groff, Weinberg

* **The SQL Complete reference :**

By Leon & Methews Leon