# Add Flight using JDBC

***Zaro Flight System*** wants to automate the process in their organization. As a start up, they need to automate the flight management system. Help them to develop this application.

You are provided with a public class Flight with following private attribute :

int flightId

String source

String destination

int noOfSeats

double flightFare

Appropriate setter and getter are written.

A public 5 argument constructor with arguments – flightId, source, destination, noOfSeats and flightFare is also provided.

Create a class FlightManagementSystem which has the following method. Use Database for manipulation.

**public boolean addFlight(Flight flightObj)** - This method should accept a flight object and add that flight details into the database. If flight details are added successfully, return true. Else, return false.

To connect to the database you are provided with database.properties file and DB.java file.

The flight table is already created at the backend. The structure of flight table is:

| Column Name | Datatype |
|---|---|
| flightId | int |
| source | varchar2(30) |
| destination | varchar2(30) |
| noofseats | int |
| flightfare | number(8,2) |

Create a class Main which has main method to perform the above operation.

In main method,

When **addFlight** method is invoked and if added successfully, print “Flight details added successfully” else print “Addition not done”.

# DB.java

```
 1
 2
 3 import java.io.FileInputStream;
 4 import java.io.IOException;
 5 import java.sql.Connection;
 6 import java.sql.DriverManager;
 7 import java.sql.SQLException;
 8 import java.util.Properties;
 9
10 public class DB {
11
12         private static Connection con = null;
13         private static Properties props = new Properties();
14
15
16     //ENSURE YOU DON'T CHANGE THE BELOW CODE WHEN YOU SUBMIT
17         public static Connection getConnection() throws ClassNotFoundException, SQLException {
18             try{
19
20                                         FileInputStream fis = null;
21                                         fis = new FileInputStream("database.properties");
22                                         props.load(fis);
23
24                                         // load the Driver Class
25                                         Class.forName(props.getProperty("DB_DRIVER_CLASS"));
26
27                                         // create the connection now
28           con =
DriverManager.getConnection(props.getProperty("DB_URL"),props.getProperty("DB_USERNAME"),props.getPr
operty("DB_PASSWORD"));
29             }
30             catch(IOException e){
31                 e.printStackTrace();
32             }
33                         return con;
34         }
35 }
36
```