Project- Airline Ticketing System

**PROJECT FOLDER NAME:** airlineticketing.zip

**Server needed**: APACHE TOMCAT 7.0.34

**Java Development Kit needed:** jdk version 6 or 7

**Database Server used:** MongoDB, MySQL, Twitter Server

**Technologies used:** Servlets, JSP, Ajax, JavaScript, Twitter API

**Total Lines of Code:**

**CSS:   
JavaScript:**   
**HTML:**   
**JAVA Servlets and JSP:**

**Python Script:**

**Total:**

**Instructions to deploy the project folder and run the application in the computer system**

1. Unzip “airlineticketing.zip” and extract the files to “webapps” folder in “apache tomcat7.0.34” folder in your computer’s storage.

2. In command prompt, go to the “airlineticketing” folder by typing the path and set the environment variables for the web application, by typing the command “env-setup-for-tomcat\_backup.bat”

3. Once the path variables are set, come out of the airlineticketing folder and in the command prompt set the path to apache –tomcat-7.0.34 ->bin.

4. Enter the command “startup.bat” to start the tomcat server.

5. Open another command prompt window and set the path to C:\Program Files\MongoDB\Server\3.2\bin and enter the command “mongod”.

6. Open another command prompt and set the path to C:\Program Files\MongoDB\Server\3.2\bin and enter “mongo.exe”

7. Now as both the Tomcat server and the MongoDB server are up and running, open the web-browser and type <http://localhost/proj/index.jsp> to start running the project.

8. To view any of the tables present inside the MySQL, open the MySQL workbench and enter the username and password as “root” and “root”. Then type “BlueWings\_Db” and “select \*from tablename” to view any of the data present inside these tables.

9. All the code needed for the MySQL and MongoDB databases is present in the *MySqlDataStoreUtilities.java* and *MongoDbDataStoreUtilities.java* file. The *web.xml* file contains the list of all the servlets implemented for this application. The python script is also present in a separate python script file.

**Note:**

Name of the database for Project: **BlueWings\_Db**

Instructions to run the “airlineticketing” web application

1. The homepage (index.jsp) contains the login and registration feature, the quick search feature using which a user can search for flights and the “About Us” column giving the introduction of the airline
2. At the top, under the logo “BlueWings” logo, there are 4 links

* Home- This link will navigate the user to the homepage of the application.
* Flights & Deals- This contains a list of all the flight specific special deals and discounts available.
* MyBookings- User can view the entire list of flight bookings done by them with an option to cancel any of it.
* Information & Services- Contains a list of all the services provided.
* Contact Us- Gives a short description as to how the user can contact the Blue Wings office in case of any issues with bookings, etc.
* LOGIN-For existing users to login into the website using their Username and Password. Once the user logs in, he/she can logout by clicking the LOGOUT option.

1. The Quick Search text box allows users to search for any flight based on the source and the destination. It’s an auto-complete search box i.e. if the user types in “g” in the search box will display the related flights.
2. The user can select any one of the items listed below to go to that particular flight and either book the flight or get more information about similar kind of flights, etc. implemented using Ajax technology.
3. The features which are implemented are:

* *Home*-Link to the home page.
* *About-Us*-A short description of Blue Wings airline and the various services provided.
* *Reviews*- This page allows the user to view the reviews of the flights he/she intends to book as well as the option to write reviews and give the ratings of the flights for which he/she has travelled. This is implemented using the Mongo DB server.
* *Bookings*-This page helps the user to book any flight.
* *Book Flight*-This functionality lets the user to book any flight by viewing the current prices and with an option to pay via debit or credit card.
* *Cancel Booking*-This functionality allows the user to cancel any booking at any time and receive a confirmation email for it.
* *View Booking*-This functionality allows the user to view the list of the flights booked by him/her and of any cancellations done previously.

1. The *Flight Deals* Page contains the options to view any holiday special deals and book the flights accordingly and also compare these deals for any competitive pricing so that the user can save up on the total amount.
2. The user has to register itself in order to book any flight by using *Registration* and he/she can view any of the flights without logging in.
3. Once the user clicks on any of the flight option for booking by specifying certain fields like the names of the passengers, etc. he/she has the option to remove these entries as well.
4. The user will have options to either checkout by completing the payment or to do log out without booking any flight.
5. Once the user clicks on the payment, there will be no option to modify the booking.
6. Once the user goes to the payment page, he/she can pay via credit or debit card and with an option to print the flight itinerary.
7. The user also has the options to view about any flight deals for which users must have tweeted from there twitter accounts. The user has the option to view these exact tweets as well which is implemented using python script.
8. The user has the option to click on the “Trending” tab and view the list of either the most preferred top 5 flights or the most popular flight deals, etc.

Administrator Login Page

**Username:** “admin” & **Password**: “admin”

1. The administrator has the option to add flights, modify flights and delete flights.
2. In the homepage of the administrator, there is an option for creating user accounts, updating user account details, and updating the customer bookings, view the flight reviews, etc.