

## Film Ticket Booking System

### 1) Features to Implement:

1. Add Ticket – Allows users to add a new ticket (TicketID, MovieName, ShowTime, SeatNumber, Price).
2. View Tickets – Displays a list of all tickets.
3. Update Ticket – Provides an interface to modify ticket details.
4. Delete Ticket – Allows users to remove a ticket record.

### 2) Project Folder Structure:

Folder	Contents
src/com.film.dao/	TicketDAO.java, TicketDAOImpl.java
src/com.film.model/	Ticket.java
src/com.film.servlet/	AddTicketServlet.java, ViewTicketServlet.java, UpdateTicketServlet.java, DeleteTicketServlet.java
src/com.film.util/	DBConnection.java
WebContent/view/	add_ticket.jsp, view_tickets.jsp, update_ticket.jsp, index.jsp
WebContent/css/	styles.css

### 3) Database Schema:

Column Name	Data Type	Constraints
ticketid	INT	PRIMARY KEY, AUTO_INCREMENT
moviename	VARCHAR(100)	NOT NULL
showtime	VARCHAR(50)	NOT NULL
seatnumber	VARCHAR(10)	NOT NULL
price	DOUBLE	NOT NULL

### CODE:

```
create table TicketBooking(  
ticketId INT PRIMARY KEY AUTO_INCREMENT,  
movieName varchar(100) NOT NULL,  
showTime VARCHAR(50) NOT NULL,  
seatNumber varchar(10) NOT NULL,  
price double not null  
);
```

**package com.film.dao;**

#### TicketDAO.java

```
import java.sql.SQLException;  
import java.util.List;  
import com.film.model.Ticket;  
public interface TicketDAO {  
    public abstract boolean add(Ticket t) throws SQLException;  
    public abstract boolean update(Ticket t);  
    public abstract boolean delete(int t);  
    public abstract List<Ticket> view();  
}
```

## TicketDAOImpl.jav

```
package com.film.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import com.film.model.Ticket;
import com.film.util.DBUtil;
public class TicketDAOImpl implements TicketDAO {
    private static Connection con;
    private static PreparedStatement pst;
    public boolean add(Ticket t) {
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement("insert into TicketBooking
(movieName,showTime,seatNumber,price) values (?,?,,?)");

        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            pst.setString(1, t.getMovieName());
            pst.setString(2, t.getShowTime());
            pst.setString(3, t.getSeatNumber());
            pst.setDouble(4, t.getPrice());
        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            if(pst.executeUpdate(>0)
                return true;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean update(Ticket t) {
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement("UPDATE TicketBooking SET movieName = ?, showTime = ?,
seatNumber = ?, price = ? WHERE ticketId = ?");
            pst.setString(1, t.getMovieName());
            pst.setString(2, t.getShowTime());
            pst.setString(3, t.getSeatNumber());
            pst.setDouble(4, t.getPrice());
            pst.setInt(5, t.getTicketId());
            return pst.executeUpdate(> 0);
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // Close resources
        }
    }
}
```

```

        try {
            if (pst != null) pst.close();
            if (con != null) con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return false;
}

public boolean delete(int ticketId) {
    try {
        con = DBUtil.getConnection();
        pst = con.prepareStatement("delete from TicketBooking where ticketId = ?");

        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            pst.setInt(1, ticketId);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            if(pst.executeUpdate()>0)
                return true;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public List<Ticket> view(){
        List<Ticket> list = new ArrayList<Ticket>();
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement("select * from TicketBooking");
            ResultSet rs = pst.executeQuery();
            while(rs.next()) {
                Ticket t = new
Ticket(rs.getInt("ticketId"),rs.getString("movieName"),rs.getString("showTime"),rs.getString("seatNumber"
),rs.getFloat("price"));

                list.add(t);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return list;
    }
}
};

```

**package com.film.model;**

**Ticket.java**

```

public class Ticket {
    int ticketId;
    String movieName;

```

```

String showTime;
String seatNumber;
double price;
public Ticket(int ticketId, String movieName, String showTime, String seatNumber, double price) {
    super();
    this.ticketId = ticketId;
    this.movieName = movieName;
    this.showTime = showTime;
    this.seatNumber = seatNumber;
    this.price = price;
}
public int getTicketId() {
    return ticketId;
}
public void setTicketId(int ticketId) {
    this.ticketId = ticketId;
}
public String getMovieName() {
    return movieName;
}
public void setMovieName(String movieName) {
    this.movieName = movieName;
}
public String getShowTime() {
    return showTime;
}
public void setShowTime(String showTime) {
    this.showTime = showTime;
}
public String getSeatNumber() {
    return seatNumber;
}
public void setSeatNumber(String seatNumber) {
    this.seatNumber = seatNumber;
}
public double getPrice() {
    return price;
}
public void setPrice(double price) {
    this.price = price;
}
@Override
public String toString() {
    return "Ticket [ticketId=" + ticketId + ", movieName=" + movieName + ", showTime=" +
showTime + ", seatNumber="
        + seatNumber + ", price=" + price + "]";
}
}

```

**package com.film.servlet;**

**AddTicketServlet.java**

```

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;

```

```

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import com.film.dao.TicketDAOImpl;
import com.film.model.Ticket;
public class AddTicketServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        int ticketId = Integer.parseInt( request.getParameter("ticketId"));
        String movieName = request.getParameter("movieName");
        String showTime = request.getParameter("showTime");
        String seatNumber= request.getParameter("seatNumber");
        double price = Float.parseFloat(request.getParameter("price"));
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Ticket t = new Ticket(ticketId,movieName,showTime,seatNumber,price);
        TicketDAOImpl dao = new TicketDAOImpl();
        if(dao.add(t)) {
            out.print("<h3>Your Ticket is Booked!</h3>");
        }
        else {
            out.print("<h3>Your Ticket is not Booked!</h3>");
        }
    }
}

```

### DeleteTicketServlet.java

```

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import com.film.dao.TicketDAOImpl;
import com.film.model.Ticket;
public class DeleteTicketServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        int ticketId = Integer.parseInt( request.getParameter("ticketId"));
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        TicketDAOImpl dao = new TicketDAOImpl();
        if(dao.delete(ticketId)) {
            out.print("<h3>Your Ticket is deleted!</h3>");
        }
        else {
            out.print("<h3>Your Ticket is not deleted!</h3>");
        }
    }
}

```

## **UpdateTicketServlet.java**

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import com.film.dao.TicketDAOImpl;
import com.film.model.Ticket;
public class UpdateTicketServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        int ticketId = Integer.parseInt( request.getParameter("ticketId"));
        String movieName = request.getParameter("movieName");
        String showTime = request.getParameter("showTime");
        String seatNumber= request.getParameter("seatNumber");
        double price = Float.parseFloat(request.getParameter("price"));
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Ticket t = new Ticket(ticketId,movieName,showTime,seatNumber,price);
        TicketDAOImpl dao = new TicketDAOImpl();
        if(dao.update(t)) {
            out.print("<h3>Your Ticket is Updated!</h3>");
        }
        else {
            out.print("<h3>Your Ticket is not Updated!</h3>");
        }
    }
}
```

## **ViewTicketServlet.java**

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
public class ViewTicketServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    }
}
```

## **package com.film.util;**

### **DBUtil.java**

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class DBUtil {
    private static Connection con;
```

```

        public static Connection getConnection() throws SQLException {
            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
                con = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/FlimTicketBooking?useSSL=false&serverTimezone=UTC","root","TToo88*9
9^");
            }

            catch(ClassNotFoundException e) {
                System.out.println(e);
            }
            return con;
        }
    }

```

### addTicket.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" href="../css/styles.css">
</head>
<body>
<form action="../AddTicketServlet" method="post">
    <fieldset>
        <legend><h2>Add Ticket</h2></legend>
        <table align='center' border="0">
            <tr>
                <td>TicketId:</td>
                <td><input type="number" name="ticketId" /></td>
            </tr>
            <tr>
                <td>MovieName:</td>
                <td><input type="text" name="movieName" /></td>
            </tr>
            <tr>
                <td>ShowTime:</td>
                <td><input type="text" name="showTime" /></td>
            </tr>
            <tr>
                <td>SeatNumber:</td>
                <td><input type="text" name="seatNumber" /></td>
            </tr>
            <tr>
                <td>Price:</td>
                <td><input type="number" name="price" /></td>
            </tr>
        </table> <br>
        <input type="submit" name="option" value="add"/>
    </fieldset>
</form>
</body>
</html>

```

### deleteTicket.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
<form action="../DeleteTicketServlet" method="post">
    <fieldset>
        <legend><h2>Delete Ticket</h2></legend>
        <table align='center' border="0">
            <tr>
                <td>TicketId:</td>
                <td><input type="number" name="ticketId" /></td>
            </tr>
        </table> <br>
        <input type="submit" name="option" value="delete"/>
    </fieldset>
</form>
</body>
</html>
```

### index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <h1>Ticket Booking System</h1>
    <div class="btns">
        <a href="addTicket.jsp">add</a>
        <a href="update.jsp">update</a>
        <a href="viewTicket.jsp">view</a>
        <a href="deleteTicket.jsp">delete</a>
    </div>
</body>
</html>
```

### update.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
```



```

<title>Insert title here</title>
  <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <form action="../UpdateTicketServlet" method="post">
        <fieldset>
            <legend><h2>Update Ticket</h2></legend>
            <table align='center' border="0">
                <tr>
                    <td>TicketId:</td>
                    <td><input type="number" name="ticketId" /></td>
                </tr>
                <tr>
                    <td>MovieName:</td>
                    <td><input type="text" name="movieName" /></td>
                </tr>
                <tr>
                    <td>ShowTime:</td>
                    <td><input type="text" name="showTime" /></td>
                </tr>
                <tr>
                    <td>SeatNumber:</td>
                    <td><input type="text" name="seatNumber" /></td>
                </tr>
                <tr>
                    <td>Price:</td>
                    <td><input type="number" name="price" /></td>
                </tr>
            </table> <br>
            <input type="submit" name="option" value="update"/>
        </fieldset>
    </form>
</body>
</html>

```

### viewTicket.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.List" %>
<%@ page import="com.film.model.Ticket" %>
<%@ page import="com.film.dao.TicketDAOImpl" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>View Tickets</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <div class="view-container">
        <h1>All Tickets</h1>
        <a href="index.jsp" class="back-link">Back to Home</a>
        <table class="ticket-table">
            <thead>
                <tr>
                    <th>Ticket ID</th>

```

```

        <th>Movie Name</th>
        <th>Show Time</th>
        <th>Seat Number</th>
        <th>Price</th>
    </tr>
</thead>
<tbody>
    <%
TicketDAOImpl dao = new TicketDAOImpl();
List<Ticket> tickets = dao.view();
if(tickets != null && !tickets.isEmpty()) {
    for(Ticket ticket : tickets) {
    %>
        <tr>
            <td><%= ticket.getTicketId() %></td>
            <td><%= ticket.getMovieName() %></td>
            <td><%= ticket.getShowTime() %></td>
            <td><%= ticket.getSeatNumber() %></td>
            <td>$<%= String.format("%.2f", ticket.getPrice()) %></td>
        </tr>
    <%
    }
    } else {
    %>
        <tr>
            <td colspan="5" class="no-data">No ticket data available</td>
        </tr>
    <% } %>
</tbody>
</table>
</div>
</body>
</html>

```

### styles.css

```

/* styles.css */
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    text-align: center;
}
.view-container {
    width: 80%;
    margin: 20px auto;
    background: white;
    padding: 15px;
    border-radius: 5px;
}
h1 {
    color: #333;
}
a {
    display: inline-block;
    margin-top: 10px;
    border-radius: 5px;
}

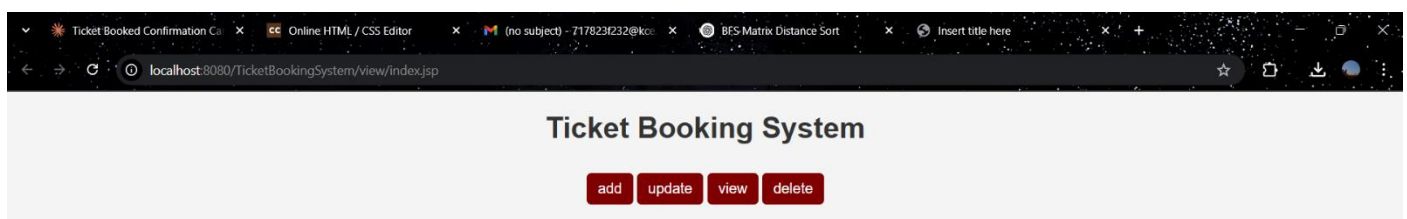
```

```

padding: 8px 12px;
background: #800000;
color: white;
text-decoration: none;
}
.ticket-table {
width: 100%;
border-collapse: collapse;
margin-top: 10px;
}
.ticket-table th,
.ticket-table td {
border: 1px solid #ccc;
padding: 8px;
}
.ticket-table thead {
background-color: #800000;
color: white;
}
.ticket-table tbody tr:nth-child(even) {
background: #f9f9f9;
}
.back-link {
display: inline-block;
margin-top: 10px;
padding: 8px 12px;
background: #800000;
color: white;
text-decoration: none;
}
.back-link:hover {
background: #0056b3;
}
fieldset{
width: 80%;
margin: 20px auto;
background: white;
padding: 15px;
border-radius: 5px;
}

```

## Output:



Insert title here

localhost:8080/TicketBookingSystem/view/addTicket.jsp

Add Ticket

TicketId: 1

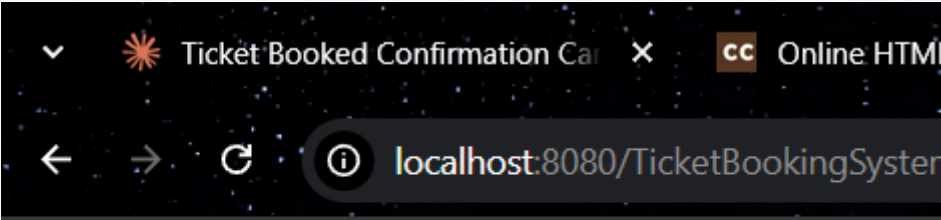
MovieName: vidamuyarchi

ShowTime: 13/5/2000 7:00PM

SeatNumber: G12

Price: 5000

add



**Your Ticket is Booked!**

View Tickets

localhost:8080/TicketBookingSystem/view/viewTicket.jsp

All Tickets

Back to Home

Ticket ID	Movie Name	Show Time	Seat Number	Price
12	vidamuyarchi	13/5/2000 7:00PM	G12	\$5000.00

Insert title here

localhost:8080/TicketBookingSystem/view/update.jsp

Update Ticket

TicketId: 12

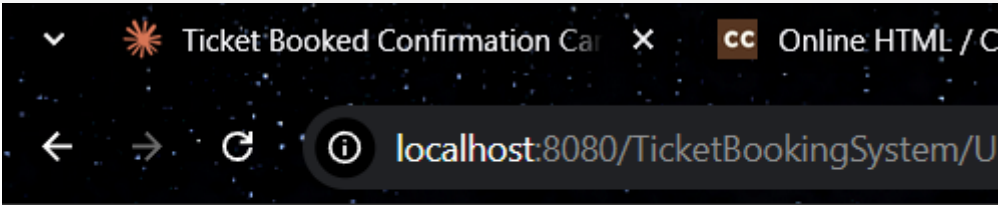
MovieName: vidamuyarchi

ShowTime: 13/2/2000 7:00PM

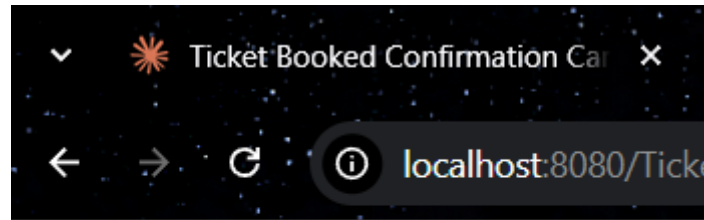
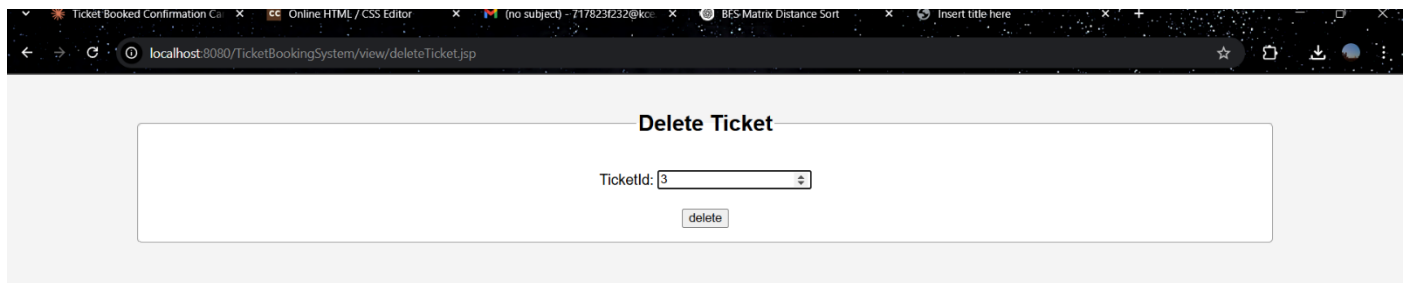
SeatNumber: L44

Price: 9999

update



**Your Ticket is Updated!**



**Your Ticket is deleted!**