

Install JDK 17 or newer.

Use IntelliJ IDEA, Eclipse, or NetBeans.

Set up a new Java project and add MySQL JDBC driver to the classpath (mysql-connector-java-X.X.X.jar).

Verify with a simple Hello World or DB connection test.

2. Define the Project Structure (1 Mark)

less

Copy

Edit

AirlineManagementSystem/

⌕

⌕ ⌕ src/

⌕ ⌕ ⌕ model/ // JavaBeans (e.g., Flight.java, Passenger.java)

⌕ ⌕ ⌕ dao/ // Data Access classes

⌕ ⌕ ⌕ db/ // DB Connection class

⌕ ⌕ ⌕ ui/ // UI (Swing/JavaFX)

⌕ ⌕ ⌕ Main.java // Entry point

⌕

⌕ ⌕ lib/ // mysql-connector.jar

3. Design the Database Schema (1 Mark)

sql

Copy

Edit

```
CREATE DATABASE airline;
```

```
USE airline;
```

```
CREATE TABLE flights (
```

```
    flight_id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    flight_number VARCHAR(10),
```

departure VARCHAR(50),

destination VARCHAR(50),

departure_time DATETIME,

arrival_time DATETIME,

seats_available INT

);

CREATE TABLE passengers (

passenger_id INT PRIMARY KEY AUTO_INCREMENT,

name VARCHAR(100),

email VARCHAR(100),

passport_number VARCHAR(20)

);

CREATE TABLE bookings (

booking_id INT PRIMARY KEY AUTO_INCREMENT,

flight_id INT,

passenger_id INT,

booking_date DATE,

FOREIGN KEY (flight_id) REFERENCES flights(flight_id),

FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id)

);

4. Create a MySQL Table (1 Mark)

Execute the above SQL schema in MySQL Workbench or phpMyAdmin.

5. Implement JDBC for Database Connectivity (3 Marks)

DBConnection.java

java

Copy

Edit

```
package db;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
public class DBConnection {
```

```
    private static final String URL = "jdbc:mysql://localhost:3306/airline";
```

```
    private static final String USER = "root";
```

```
    private static final String PASSWORD = "your_password";
```

```
    public static Connection getConnection() throws SQLException {
```

```
        return DriverManager.getConnection(URL, USER, PASSWORD);
```

```
}
```

```
}
```

Test it in Main.java:

```
java
```

Copy

Edit

```
public static void main(String[] args) {
```

```
    try {
```

```
        Connection conn = DBConnection.getConnection();
```

```
        System.out.println("Connected!");
```



```
} catch (SQLException e) {
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

6. Create Model and DAO Classes (3 Marks)

Model - Flight.java

java

Copy

Edit

```
package model;
```

```
public class Flight {  
  
    private int flightId;  
  
    private String flightNumber;  
  
    private String departure;  
  
    private String destination;  
  
  
    // Getters and setters  
  
}
```

DAO - FlightDAO.java

java

Copy

Edit

```
package dao;
```

```
import db.DBConnection;
```

```
import model.Flight;
```

```
import java.sql.*;
```

```
import java.util.*;
```

```
public class FlightDAO {

    public List<Flight> getAllFlights() {

        List<Flight> flights = new ArrayList<>();

        try (Connection conn = DBConnection.getConnection();

            Statement stmt = conn.createStatement();

            ResultSet rs = stmt.executeQuery("SELECT * FROM flights")) {

            while (rs.next()) {

                Flight flight = new Flight();

                flight.setFlightId(rs.getInt("flight_id"));

                flight.setFlightNumber(rs.getString("flight_number"));

            }

        }

    }

}
```

```
        flight.setDeparture(rs.getString("departure"));

        flight.setDestination(rs.getString("destination"));

        flights.add(flight);

    }

} catch (SQLException e) {

    e.printStackTrace();

}

return flights;

}

}
```

7. Aesthetics and Visual Appeal of the UI (4 Marks)

Java Swing UI Example:

Use a JFrame with custom colors, fonts, and icons.

Load flight data into a JTable.

java

Copy

Edit

```
JFrame frame = new JFrame("Airline Management System");
```

```
frame.setSize(800, 600);
```

```
frame.setLayout(new BorderLayout());
```

```
JLabel heading = new JLabel("Flight Dashboard", SwingConstants.CENTER);
```

```
heading.setFont(new Font("Arial", Font.BOLD, 26));
```

```
heading.setForeground(Color.BLUE);
```

```
frame.add(heading, BorderLayout.NORTH);
```

Design Considerations:

Consistent theme

Application icon

Modern-looking components

8. Component Placement and Alignment (2 Marks)

Use layout managers:

GridBagLayout for forms

BorderLayout or BoxLayout for panels

Example:

java

Copy

Edit

```
JPanel formPanel = new JPanel(new GridBagLayout());
```

```
formPanel.add(new JLabel("Flight No:"));
```

```
formPanel.add(new JTextField(10));
```

9. Responsiveness and Accessibility (2 Marks)

Use JScrollPane for tables/forms.

Keyboard navigation (Tab/Enter).

Accessible fonts and labels.

Example:

java

Copy

Edit

```
JTable table = new JTable(data, columnNames);
```

```
JScrollPane scrollPane = new JScrollPane(table);
```

```
frame.add(scrollPane, BorderLayout.CENTER);
```