**Leave Management System**

**Step 1: Set Up Your Java Project**

* Open Eclipse or IntelliJ IDEA.
* Create a Dynamic Web Project (if using Eclipse) or a Maven Web App (if using IntelliJ).
* Name the project "Bus Reservation System**".**

**Step 2: Configure Tomcat Server**

Download and install Apache Tomcat (if not already installed).

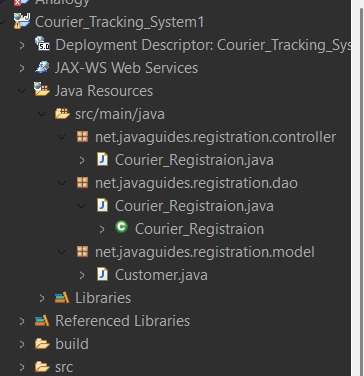
In Eclipse, go to:

* Pgsql

Window → Preferences → Server → Runtime Environments → Add → Apache Tomcat

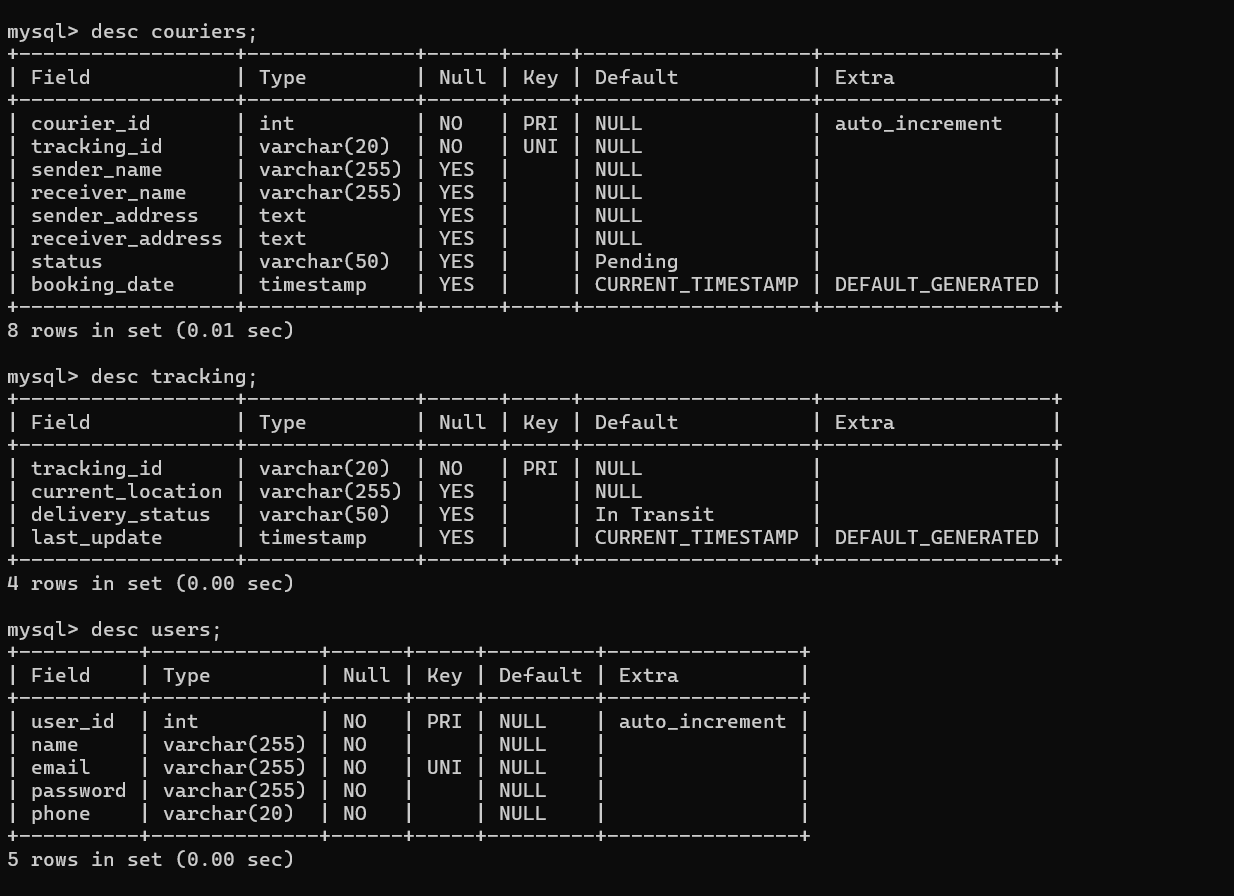
**Step 3: Set Up the Project Structure**

Create the following directories in your project:



**Step 4: Create the Database**

Open MySQL Workbench or any database tool and create a database.



Step 5: Create the Database Connection

Inside the util/ package, create a class Customer\_Regeistration\_Connection.java.

package net.javaguides.registration.dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import net.javaguides.registration.model.Customer;

public class Customer\_Registration {

public int registerCustomer(Customer customer) throws ClassNotFoundException {

String INSERT\_USERS\_SQL = "INSERT INTO courier (user\_id, name, email, password, phone) VALUES (?, ?, ?, ?, ?);";

int result = 0;

Class.forName("com.mysql.cj.jdbc.Driver"); // Updated driver

try (Connection cn = DriverManager.getConnection("jdbc: mysql://localhost:3306/courier", "root", "Shubham@05");

PreparedStatement preparedStatement = cn.prepareStatement(INSERT\_USERS\_SQL)) {

preparedStatement.setInt(1, customer.getUser\_Id());

preparedStatement.setString(2, customer.getName());

preparedStatement.setString(3, customer.getEmail());

preparedStatement.setString(4, customer.getPassword());

preparedStatement.setLong(5, customer.getPhone()); // Changed to String

System.out.println(preparedStatement);

result = preparedStatement.executeUpdate();

} catch (SQLException e) {

printSQLException(e);

}

return result;

}

private void printSQLException(SQLException ex) {

for (Throwable e : ex) {

if (e instanceof SQLException) {

e.printStackTrace(System.err);

}

}

}

}

Step 6: Create the Model Classes

Inside model/, create Customer.java.

package net.javaguides.registration.model;

public class Customer {

private int user\_Id;

private int phone;

private String name;

private String email;

private String password;

public int getUser\_Id() {

return user\_Id;

}

public void setUser\_Id(int user\_Id) {

this.user\_Id = user\_Id;

}

public int getPhone() {

return phone;

}

public void setPhone(int phone) {

this.phone = phone;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

**Step 7: Create the Servlets (Controllers)**

Inside controller/, create Customer\_Registration(servlet).java.

package net.javaguides.registration.controller;

import java.io.IOException;

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 net.javaguides.registration.model.Customer;

@WebServlet("/Courier\_Registraion")

public class Courier\_Registraion extends HttpServlet {

private static final long serialVersionUID = 1L.

public Courier\_Registraion() {

super ();

}

// Handling GET requests - Display the registration form

protected void do Get (HttpServletRequest request, HttpServletResponse response) throws Servlet Exception, IOException {

// Forward the request to the registration page

}

// Handling POST requests - Register the customer

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// Get form data

try {

int id = Integer.parseInt(request.getParameter("User\_id”);

String name = request.getParameter("name");

String email = request.getParameter("email");

String password = request.getParameter("password");

int phone = Integer.parseInt(request.getParameter("phone"));

// Creating Customer object

Customer cu = new Customer ();

cu.setUser\_Id(id);

cu.setName(name);

cu.setEmail(email);

cu.setPassword(password);

cu.setPhone(phone);

// Register customer in the database (you can implement this logic)

// Example: Customer\_Registration.registerCustomer(cu);

request.setAttribute("message", "Customer registered successfully!");

} catch (Exception e) {

// In case of error, set the error message

request.setAttribute("error", "Error: " + e.getMessage());

}

// Forwarding the request back to the registration page (with message)

}

}

Step 8: Create the JSP Views

Customer\_Registration.jsp

<%@ **page** language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>Customer Registration</**title**>

</**head**>

<**body**>

<**h2**>Customer Registration Form</**h2**>

<**form** action=*"Gym\_Registraion"* method=*"post"*>

<**label** for=*"User\_id"*>User ID:</**label**>

<**input** type=*"number"* id=*"user\_id"* name=*"user\_id"* required><**br**><**br**>

<**label** for=*"name"*>Name:</**label**>

<**input** type=*"text"* id=*"name"* name=*"name"* required><**br**><**br**>

<**label** for=*"email"*>Email:</**label**>

<**input** type=*"email"* id=*"email"* name=*"email"* required><**br**><**br**>

<**label** for=*"password"*>Password:</**label**>

<**input** type=*"password"* id=*"password"* name=*"password"* required><**br**><**br**>

<**label** for=*"phone"*>Phone:</**label**>

<**input** type=*"number"* id=*"phone"* name=*"phone"* required><**br**><**br**>

<**button** type=*"submit"*>Register</**button**>

</**form**>

</**body**>

</**html**>

**Step 11: Run the Project**

Right-click the project → Run As → Run on Server (Eclipse) or Deploy to Tomcat (IntelliJ).

Open <http://localhost:8080/Bus>ReservationSystem /track.jsp.

Enter a tracking number and check the status.