ATTENDANCE MANAGEMENT SYSTEM

Create an attendance management system for universitylevel graduates using Java, following the MVC architecture,

And employing Maven as a build tool. This system utilizes

MySQL or another database for backend data management

And deploys on a Tomcat server to allow live access and

Usage.

Src

├── main

│ ├── java

│ │ ├── model

│ │ ├── view

│ │ └── controller

│ ├── resources

│ └── webapp

│ ├── WEB-INF

│ │ └── web.xml

│ ├── css

│ ├── js

│ └── jsp

└── test

Xml;

<dependencies>

<!—MySQL Connector 🡪

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.34</version>

</dependency>

<!—Servlet API 🡪

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!—JSTL 🡪

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

<build>

<plugins>

<!—Tomcat Plugin 🡪

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

</plugin>

</plugins>

</build>

Sql;

CREATE TABLE students (

Id INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Course VARCHAR(100)

);

CREATE TABLE attendance (

Id INT AUTO\_INCREMENT PRIMARY KEY,

Student\_id INT,

Date DATE,

Status VARCHAR(10),

FOREIGN KEY (student\_id) REFERENCES students(id)

);

Java;

Package model;

Public class Student {

Private int id;

Private String name;

Private String email;

Private String course;

// Getters and Setters

}

Java;

Package model;

Import java.sql.\*;

Import java.util.\*;

Public class StudentDAO {

Private Connection connection;

Public StudentDAO(Connection connection) {

This.connection = connection;

}

Public List<Student> getAllStudents() throws SQLException {

List<Student> students = new ArrayList<>();

String query = “SELECT \* FROM students”;

PreparedStatement statement = connection.prepareStatement(query);

ResultSet rs = statement.executeQuery();

While (rs.next()) {

Student student = new Student();

Student.setId(rs.getInt(“id”));

Student.setName(rs.getString(“name”));

Student.setEmail(rs.getString(“email”));

Student.setCourse(rs.getString(“course”));

Students.add(student);

}

Return students;

}

Public void addStudent(Student student) throws SQLException {

String query = “INSERT INTO students (name, email, course) VALUES (?, ?, ?)”;

PreparedStatement statement = connection.prepareStatement(query);

Statement.setString(1, student.getName());

Statement.setString(2, student.getEmail());

Statement.setString(3, student.getCourse());

Statement.executeUpdate();

}

}

Jsp;

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

<%@ taglib uri=<http://java.sun.com/jsp/jstl/core> prefix=”c” %>

<!DOCTYPE html>

<html>

<head>

<title>Attendance Management</title>

</head>

<body>

<h1>Student List</h1>

<table border=”1”>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Course</th>

</tr>

<c:forEach var=”student” items=”${students}”>

<tr>

<td>${student.id}</td>

<td>${student.name}</td>

<td>${student.email}</td>

<td>${student.course}</td>

</tr>

</c:forEach>

</table>

</body>

</html>

Java;

Package controller;

Import model.Student;

Import model.StudentDAO;

Import javax.servlet.\*;

Import javax.servlet.http.\*;

Import java.io.IOException;

Import java.sql.\*;

Import java.util.\*;

Public class StudentController extends HttpServlet {

Private StudentDAO studentDAO;

@Override

Public void init() {

Try {

Connection connection = DriverManager.getConnection(

“jdbc:mysql://localhost:3306/attendance\_management”, “root”, “password”);

studentDAO = new StudentDAO(connection);

} catch (SQLException e) {

Throw new RuntimeException€;

}

}

@Override

Protected void doGet(HttpServletRequest request, HttpServletResponse response)

Throws ServletException, IOException {

Try {

List<Student> students = studentDAO.getAllStudents();

Request.setAttribute(“students”, students);

RequestDispatcher dispatcher = request.getRequestDispatcher(“/jsp/index.jsp”);

Dispatcher.forward(request, response);

} catch (SQLException e) {

Throw new ServletException€;

}

}

}

Xml;

<web-app xmlns=<http://java.sun.com/xml/ns/javaee> version=”3.1”>

<servlet>

<servlet-name>StudentController</servlet-name>

<servlet-class>controller.StudentController</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>StudentController</servlet-name>

<url-pattern>/students</url-pattern>

</servlet-mapping>

</web-app>

WORD COUNTER

A simple project for beginners is good to start. It can be built

Using Swing in Java. Here, the application tells you the no of

Words, the entered paragraph has.

Java;

Import javax.swing.\*;

Import java.awt.\*;

Import java.awt.event.ActionEvent;

Import java.awt.event.ActionListener;

Public class WordCounter {

Public static void main(String[] args) {

// Create the JFrame

JFrame frame = new JFrame(“Word Counter”);

Frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Frame.setSize(400, 300);

Frame.setLayout(new BorderLayout());

// Create a JTextArea for input

JTextArea textArea = new JTextArea();

textArea.setLineWrap(true);

textArea.setWrapStyleWord(true);

JScrollPane scrollPane = new JScrollPane(textArea);

Frame.add(scrollPane, BorderLayout.CENTER);

// Create a JPanel for the button and result

JPanel panel = new JPanel();

Panel.setLayout(new BorderLayout());

// Create a JButton to count words

JButton countButton = new JButton(“Count Words”);

Panel.add(countButton, BorderLayout.WEST);

// Create a JLabel to display the word count

JLabel wordCountLabel = new JLabel(“Word Count: 0”);

Panel.add(wordCountLabel, BorderLayout.CENTER);

Frame.add(panel, BorderLayout.SOUTH);

// Add ActionListener to the button

countButton.addActionListener(new ActionListener() {

@Override

Public void actionPerformed(ActionEvent e) {

String text = textArea.getText().trim();

If (text.isEmpty()) {

wordCountLabel.setText(“Word Count: 0”);

} else {

String[] words = text.split(<\\s+>); // Split text by spaces

wordCountLabel.setText(“Word Count: “ + words.length);

}

}

});

// Make the frame visible

Frame.setVisible(true);

}

}