**Models**

**-------------------------------------**

**Class.java**

**package** com.simplilearn.models;

**public** **class** Class {

**private** **int** id;

**private** **int** section;

**private** String teacher;

**private** String subject;

**private** String time;

**public** Class(**int** id, **int** section, String teacher, String subject, String time) {

**super**();

**this**.id = id;

**this**.section = section;

**this**.teacher = teacher;

**this**.subject = subject;

**this**.time = time;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** **int** getSection() {

**return** section;

}

**public** **void** setSection(**int** section) {

**this**.section = section;

}

**public** String getTeacher() {

**return** teacher;

}

**public** **void** setTeacher(String teacher) {

**this**.teacher = teacher;

}

**public** String getSubject() {

**return** subject;

}

**public** **void** setSubject(String subject) {

**this**.subject = subject;

}

**public** String getTime() {

**return** time;

}

**public** **void** setTime(String time) {

**this**.time = time;

}

}

**Student.java**

**package** com.simplilearn.models;

**public** **class** Student {

**private** **int** id;

**private** String fname;

**private** String lname;

**private** **int** age;

**private** **int** student\_class;

**public** Student(**int** id, String fname, String lname, **int** age, **int** student\_class) {

**super**();

**this**.id = id;

**this**.fname = fname;

**this**.lname = lname;

**this**.age = age;

**this**.student\_class = student\_class;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return** lname;

}

**public** **void** setLname(String lname) {

**this**.lname = lname;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **int** getAclass() {

**return** student\_class;

}

**public** **void** setAclass(**int** student\_class) {

**this**.student\_class = student\_class;

}

@Override

**public** String toString() {

**return** "Student [id=" + id + ", fname=" + fname + ", lname=" + lname + ", age=" + age + ", class=" + student\_class

+ "]";

}

}

**Subjects.java**

**package** com.simplilearn.models;

**public** **class** Subject {

**private** **int** id;

**private** String name;

**private** String short\_form;

**public** Subject(**int** id, String name, String short\_form ) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.short\_form = short\_form;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getShortform() {

**return** short\_form;

}

**public** **void** setShortform(String short\_form) {

**this**.short\_form = short\_form;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**Teacher.java**

**package** com.simplilearn.models;

**public** **class** Teacher {

**private** **int** id;

**private** String fname;

**private** String lname;

**private** **int** age;

**public** Teacher(**int** id, String fname, String lname, **int** age) {

**super**();

**this**.id = id;

**this**.fname = fname;

**this**.lname = lname;

**this**.age = age;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getFname() {

**return** fname;

}

**public** **void** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return** lname;

}

**public** **void** setLname(String lname) {

**this**.lname = lname;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

}

**Admin:**

**TestServlet.java**

package com.simplilearn.admin;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.annotation.Resource;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.sql.DataSource;

@WebServlet("/TestServlet")

public class TestServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

@Resource(name="jdbc\_database")

private DataSource dataSource;

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

PrintWriter out = response.getWriter();

response.setContentType("text/plain");

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "select \* from students";

statement = connection.createStatement();

results = statement.executeQuery(sql);

while(results.next()) {

String fname = results.getString("fname");

out.println(fname);

}

}

catch(Exception e) {

e.printStackTrace();

}

}

}

**AdminControllerServlet.java**

**package** com.simplilearn.admin;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.List;

**import** javax.annotation.Resource;

**import** javax.servlet.RequestDispatcher;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.Cookie;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.sql.DataSource;

**import** com.simplilearn.models.Student;

**import** com.simplilearn.models.Subject;

**import** com.simplilearn.models.Teacher;

**import** com.simplilearn.models.Class;

@WebServlet("/AdminControllerServlet")

**public** **class** AdminControllerServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** DbRetrieve dbRetrieve;

@Resource(name = "jdbc\_database")

**private** DataSource datasource;

@Override

**public** **void** init() **throws** ServletException {

**super**.init();

**try** {

dbRetrieve = **new** DbRetrieve(datasource);

}

**catch** (Exception e) {

**throw** **new** ServletException(e);

}

}

**public** AdminControllerServlet() {

**super**();

}

@Override

**protected** **void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

doGet(req, resp);

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

**try** {

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

**if** (command == **null**) {

command = "CLASSES";

}

**if** (!command.equals("LOGIN") && !getCookies(request, response) ) {

response.sendRedirect("/Administrative-Portal/login.jsp");

}

**else** {

**switch** (command) {

**case** "STUDENTS":

studentsList(request, response);

**break**;

**case** "TEACHERS":

teachersList(request, response);

**break**;

**case** "SUBJECTS":

subjectList(request, response);

**break**;

**case** "CLASSES":

classestList(request, response);

**break**;

**case** "ST\_LIST":

classStudentsList(request, response);

**break**;

**case** "LOGIN":

login(request, response);

**break**;

**default**:

classestList(request, response);

}

}

} **catch** (Exception e) {

**throw** **new** ServletException(e);

}

}

**private** **void** studentsList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Student> students = dbRetrieve.getStudents();

request.setAttribute("STUDENT\_LIST", students);

RequestDispatcher dispatcher = request.getRequestDispatcher("/list-students.jsp");

dispatcher.forward(request, response);

}

**private** **void** teachersList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Teacher> teachers = dbRetrieve.getTeachers();

request.setAttribute("TEACHERS\_LIST", teachers);

RequestDispatcher dispatcher = request.getRequestDispatcher("/teachers-list.jsp");

dispatcher.forward(request, response);

}

**private** **void** subjectList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Subject> subjects = dbRetrieve.getSubjects();

request.setAttribute("SUBJECTS\_LIST", subjects);

RequestDispatcher dispatcher = request.getRequestDispatcher("/subjects-list.jsp");

dispatcher.forward(request, response);

}

**private** **void** classestList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

List<Class> classes = dbRetrieve.getClasses();

request.setAttribute("CLASSES\_LIST", classes);

RequestDispatcher dispatcher = request.getRequestDispatcher("/classes-list.jsp");

dispatcher.forward(request, response);

}

**private** **void** login(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

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

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

**if** (username.toLowerCase().equals("admin") && password.toLowerCase().equals("admin@123")) {

Cookie cookie = **new** Cookie(username, password);

cookie.setMaxAge(86400);

response.addCookie(cookie);

classestList(request, response);

}

**else** {

RequestDispatcher dispatcher = request.getRequestDispatcher("/login.jsp");

dispatcher.forward(request, response);

}

}

**private** **void** classStudentsList(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

**int** classId = Integer.*parseInt*(request.getParameter("classId"));

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

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

List<Student> students = dbRetrieve.loadClassStudents(classId);

request.setAttribute("STUDENTS\_LIST", students);

request.setAttribute("SECTION", section);

request.setAttribute("SUBJECT", subject);

RequestDispatcher dispatcher = request.getRequestDispatcher("/class-students.jsp");

dispatcher.forward(request, response);

}

**private** **boolean** getCookies(HttpServletRequest request, HttpServletResponse response) **throws** Exception {

**boolean** check = **false**;

Cookie[] cookies = request.getCookies();

**for** (Cookie cookie : cookies) {

**if** (cookie.getName().equals("admin") && cookie.getValue().equals("admin@123")) {

check = **true**;

**break**;

}

}

**return** check;

}

}

**DbRetrieve.java**

package com.simplilearn.admin;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import javax.sql.DataSource;

import com.simplilearn.models.Student;

import com.simplilearn.models.Subject;

import com.simplilearn.models.Teacher;

import com.simplilearn.models.Class;

public class DbRetrieve {

private DataSource dataSource;

public DbRetrieve(DataSource dataSource) {

this.dataSource = dataSource;

}

public List<Student> getStudents() {

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

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM students";

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

String firstName = results.getString("fname");

String lastName = results.getString("lname");

int age = results.getInt("age");

int student\_class = results.getInt("class");

Student tempStudent = new Student(id, firstName, lastName, age, student\_class);

students.add(tempStudent);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return students;

}

public List<Teacher> getTeachers() {

List<Teacher> teachers = new ArrayList<>();

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM teachers";

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

String firstName = results.getString("fname");

String lastName = results.getString("lname");

int age = results.getInt("age");

Teacher temp = new Teacher(id, firstName, lastName, age);

teachers.add(temp);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return teachers;

}

public List<Subject> getSubjects() {

List<Subject> subjects = new ArrayList<>();

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM subjects";

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

String name = results.getString("name");

String shortform = results.getString("shortform");

Subject temp = new Subject(id, name,shortform);

subjects.add(temp);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return subjects;

}

public List<Class> getClasses() {

List<Class> classes = new ArrayList<>();

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM classes";

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

int section = results.getInt("section");

int subject = results.getInt("subject");

int teacher = results.getInt("teacher");

String time = results.getString("time");

Teacher tempTeacher = loadTeacher(teacher);

Subject tempSubject = loadSubject(subject);

String teacher\_name = tempTeacher.getFname() + " " + tempTeacher.getLname();

Class temp = new Class(id, section, teacher\_name, tempSubject.getName(), time);

classes.add(temp);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return classes;

}

public Teacher loadTeacher(int teacherId) {

Teacher theTeacher = null;

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM teachers WHERE id = " + teacherId;

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

String fname = results.getString("fname");

String lname = results.getString("lname");

int age = results.getInt("age");

theTeacher = new Teacher(id, fname, lname, age);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return theTeacher;

}

public Subject loadSubject(int subjectId) {

Subject theSubject = null;

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM subjects WHERE id = " + subjectId;

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

String name = results.getString("name");

String shortform = results.getString("shortform");

theSubject = new Subject(id, name,shortform);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return theSubject;

}

public Class loadClass(int classId) {

Class theClass = null;

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM clasess WHERE id = " + classId;

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

int section = results.getInt("section");

int subject = results.getInt("subject");

int teacher = results.getInt("teacher");

String time = results.getString("time");

Teacher tempTeacher = loadTeacher(teacher);

Subject tempSubject = loadSubject(subject);

String teacher\_name = tempTeacher.getFname() + " " + tempTeacher.getLname();

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return theClass;

}

public List<Student> loadClassStudents(int classId) {

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

Connection connection = null;

Statement statement = null;

ResultSet results = null;

try {

connection = dataSource.getConnection();

String sql = "SELECT \* FROM students WHERE class = " + classId;

statement = connection.createStatement();

results = statement.executeQuery(sql);

while (results.next()) {

int id = results.getInt("id");

String firstName = results.getString("fname");

String lastName = results.getString("lname");

int age = results.getInt("age");

int student\_class = results.getInt("class");

Student tempStudent = new Student(id, firstName, lastName, age, student\_class);

students.add(tempStudent);

}

} catch (Exception e) {

} finally {

close(connection, statement, results);

}

return students;

}

private void close(Connection connection, Statement statement, ResultSet results) {

try {

if (results != null) {

results.close();

}

if (statement != null) {

statement.close();

}

if (connection != null) {

connection.close();

}

} catch (Exception e) {

e.printStackTrace();

}

}

}