

Learner's Academy App Version-1

Project Developer: Priyadarshi Panigrahi
Email Id: priyadarshi_05@yahoo.co.in

GitHubLink

https://github.com/Priyam777/LearnersAcademyApp_Priyadarshi_Panigrahi.git

Brief Description : The project uses Servlets, JSP, MySQL database server and Hibernate to implement the specific
*The application has two parts: One for data upload to MySQL database and the other part is for report
*The admin logs in by providing the right credentials and is then directed to a page which provides the features to upload data and view the reports.

Admin login Feature:

UI

Admin Name:

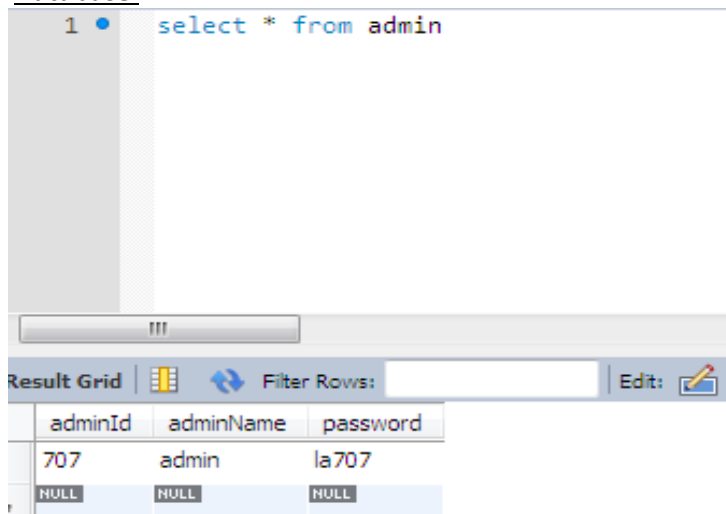
Password :

Login

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-
priyadarshi_05@yahoo.co.in

Database



The screenshot shows a database query editor with a SQL query: `select * from admin`. Below the query, there is a 'Result Grid' showing the results of the query. The grid has three columns: 'adminId', 'adminName', and 'password'. The first row contains the values '707', 'admin', and 'la707'. The second row contains 'NULL', 'NULL', and 'NULL'.

adminId	adminName	password
707	admin	la707
NULL	NULL	NULL

Code:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    HttpSession session=request.getSession();  
    String adminName=request.getParameter("adminName");  
  
    String password=request.getParameter("password");  
  
    List<Admin> adminList=LearnersAcademyDao.getAdmin();
```

```

System.out.println(adminList);
if(adminList.size()==0){
    PrintWriter out=response.getWriter();
    String message="<span style='color:red'>Invalid credentials.Try again</span>";
    session.setAttribute("message", message);
    request.getRequestDispatcher("AdminLogin.jsp").include(request, response);
    out.print(message);
}else{
    for(Admin a:adminList){
        System.out.println(a);
        if((a!=null)&& a.getAdminName().equalsIgnoreCase(adminName)&&
            a.getPassword().equalsIgnoreCase(password)){
            request.getRequestDispatcher("AdminHome.jsp").forward(request, response);

        }else{
            PrintWriter out=response.getWriter();
            String message="<span style='color:red'>Invalid credentials.Try again</span>";
            session.setAttribute("message", message);
            request.getRequestDispatcher("AdminLogin.jsp").include(request, response);
            out.print(message);
        }
    }
}

```

Admin Home Page on successful login

UI

Logged in Successfully at Sat Jun 05 00:10:24 IST 2021

Admin Utility

[Add Subjects for a given class](#)

[Add Students for a given class](#)

[Add Teachers](#)

[Add Classes](#)

[Assign class to a given Teacher](#)

[Assign subjects to a given teacher](#)

Admin Reports

[Students Report](#)

[Subjects Report](#)

[Teachers Report](#)

[Classes Report](#)

[Master Report](#)

Add Subjects for a given class

UI

Enter Class Name:

Enter Subject Name:

[Go back to Admin Home page](#)

[Add more subjects](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

On hitting the Add Subject Button

Enter Class Name:

Enter Subject Name:

[Go back to Admin Home page](#)

[Add more subjects](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

1subject added

Database:

1 • `select * from subject`

Result Grid

Filter Rows:

Edit:

	subjectId	subjectName	subjectClass_classId
▶	1	maths-10	10
	2	physics-10	10
	3	history-10	10
	4	drawing-1	1
	5	drawing-2	2
	6	english-3	3
	7	socialstudies-4	4
	8	history-5	5
	9	maths-6	6
	10	history-6	6
	11	physics-7	7
	12	history-8	8
	13	maths-9	9
	14	chemistry-9	9
	15	geography-10	10

15	geograpny-10	10
16	english-5	5

Code:

```

public static int addSubjects(String className,String subjectName){
    Session session=null;
    Transaction transaction=null;
    int count=0;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();
        Class c1=new Class();

        Query query=session.createQuery("from Class c where c.className=:className");
        query.setParameter("className", className);
        List classList=query.list();
        System.out.println(classList);

        if(classList.size()==0){
            c1.setClassName(className);
            session.save(c1);
        }
        else{
            c1=(Class)classList.get(0);

        }

        Subject s=new Subject();

        Query query1=session.createQuery("from Subject s where s.subjectName=:subjectName ");
        query1.setParameter("subjectName", subjectName);
        //query1.setParameter("className", className);
        List subjectList=query1.list();
        System.out.println(subjectList);
        if(subjectList.size()==0){
            s.setSubjectClass(c1);
            s.setSubjectName(subjectName);
            session.merge(s);
            count=1;
        }else{
            count=-1;
        }

    }catch(Exception e){

```

```

        System.out.println(e.getMessage());
    }finally{
        transaction.commit();
    }
    if(session!=null){
        session.close();
    }
    return count;
}

```

Add Students for a given class

UI

Enter Class Name:
 Enter Student Name:

[Go back to Admin Home page](#)

[Add more students](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-
 priyadarshi_05@yahoo.co.in

On hitting the Add Student button

Enter Class Name:
 Enter Student Name:

[Go back to Admin Home page](#)

[Add more students](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-
 priyadarshi_05@yahoo.co.in

11student added

Database:

1	select * from student
---	-----------------------

studentId	studentName	studentClass_classId
1	Anjali kumari	1
2	Umesh	2
3	Dang Lou	3
4	Hitesh	4
5	Remo DeCosta	5
6	Sanjana	6
7	kishore	7

8	Stalin	8
9	K Sathish	9
10	Aswini Kumar	10
11	Satnam	7
NULL	NULL	NULL

Code:

```

public static int addStudents(String className,String studentName){
    Session session=null;
    Transaction transaction=null;
    int count=0;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();
        Class c1=new Class();

        Query query=session.createQuery("from Class c where c.className=:className");
        query.setParameter("className", className);
        List classList=query.list();
        System.out.println(classList);

        if(classList.size()==0){
            c1.setClassName(className);
            session.save(c1);
        }
        else{
            c1=(Class)classList.get(0);

        }

        Student s=new Student();
        s.setStudentClass(c1);
        s.setStudentName(studentName);
        count=(int)session.save(s);

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();

        if(session!=null){
            session.close();
        }
    }
}

```

```

    }
    return count;
}
}

```

Add a class

UI

Enter Class Name:

Add Class

[Go back to Admin Home page](#)

[Add more classes](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

On hitting the Add class button

Enter Class Name:

Add Class

[Go back to Admin Home page](#)

[Add more classes](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

class added successfully

Database

1	select * from class
Filter Rows: <input type="text"/>	
Edit:	
classId	className
1	class-1
2	class-2
3	class-3
4	class-4
5	class-5
6	class-6
7	class-7
8	class-8
9	class-9
10	class-10

13	class-11
NULL	NULL

Code:

```

public static int addClasses(String className){
    Session session=null;
    Transaction transaction=null;
    int count=0;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();
        Class c1=new Class();

        Query query=session.createQuery("from Class c where c.className=:className");
        query.setParameter("className", className);
        List classList=query.list();
        System.out.println(classList);

        if(classList.size()==0){
            c1.setClassName(className);
            session.save(c1);
            count=1;
        }else count= -1;

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();

        if(session!=null){
            session.close();
        }
        return count;
    }
}

```

Add a Teacher

UI

Enter Teacher Name:

[Go back to Admin Home page](#)

[Add more Teachers](#)

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

On hitting the Add Teacher button

Enter Teacher Name:

Add Teacher

[Go back to Admin Home page](#)

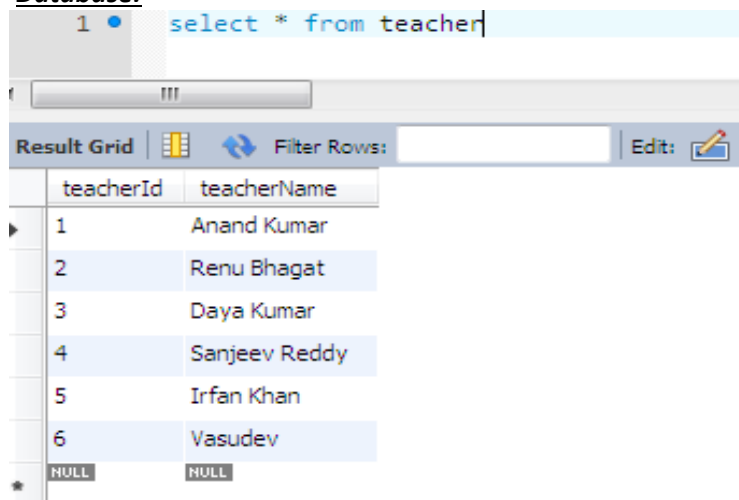
[Add more Teachers](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

1Teacher added

Database:



The screenshot shows a database management interface. At the top, a SQL query is entered: `select * from teacher`. Below the query, there is a 'Result Grid' section. It contains a table with two columns: 'teacherId' and 'teacherName'. The table lists six teachers with IDs 1 through 6. The last row shows 'NULL' for both fields. The interface also includes a 'Filter Rows' search bar and an 'Edit' button.

teacherId	teacherName
1	Anand Kumar
2	Renu Bhagat
3	Daya Kumar
4	Sanjeev Reddy
5	Irfan Khan
6	Vasudev
NULL	NULL

Code:

```
public static int addTeachers(String teacherName){
    Session session=null;
    Transaction transaction=null;
    int count=0;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();
        Teacher t1=new Teacher();

        Query query=session.createQuery("from Teacher t where t.teacherName=:teacherName");
        query.setParameter("teacherName", teacherName);
```

```

        query.setParameter("teacherName", teacherName);
        List teacherList=query.list();
        System.out.println(teacherList);

        if(teacherList.size()==0){
            t1.setTeacherName(teacherName);
            session.save(t1);
            count=1;
        }else count= -1;

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();
    }

    if(session!=null){
        session.close();
    }

    return count;
}
}

```

Assign a teacher to a class

UI

Enter Teacher Name: Vasudeva

Enter class Name: class-11

Assign class to Teacher

[Go back to Admin Home page](#)

[Add Teacher](#)

[Assign more classes to Teacher](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

On hitting the Assign class to Teacher button

Enter Teacher Name:

Enter class Name:

Assign class to Teacher

[Go back to Admin Home page](#)

[Add Teacher](#)

[Assign more classes to Teacher](#)


©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

1class assigned to teacher

Database

1 • select * from teacher_class

Result Grid | Filter Rows: | Edit: 

Teacher_teacherId	classList_classId
2	9
2	10
2	7
1	10
3	5
3	1
5	9
5	6
4	2
4	7
6	13
NULL	NULL

★

Code:

```

public static int assignTeacherClass(String teacherName,String className){
    Session session=null;
    Transaction transaction=null;
    int count=0;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();
        Class c1=new Class();

        Query query=session.createQuery("from Class c where c.className=:className");
        query.setParameter("className", className);
        List classList=query.list();
        System.out.println(classList);

        if(classList.size()==0){
            c1.setClassName(className);
            session.save(c1);
        }
        else{
            c1=(Class)classList.get(0);
        }
        System.out.println(c1);
        Teacher t1=new Teacher();
    }
}

```

```

Query query1=session.createQuery("from Teacher t where t.teacherName=:teacherName");
query1.setParameter("teacherName", teacherName);
List teacherList=query1.list();
System.out.println(teacherList);

if(teacherList.size()==0){
    //t1.setTeacherName(teacherName);
    //session.save(t1);
    count=-1;
}else {
    t1=(Teacher)teacherList.get(0);

    if(t1.getClassList().contains(c1)){
        count=0;
    }
    else{
        t1.getClassList().add(c1);
        session.merge(t1);
        count=1;
    }
}

}

}catch(Exception e){
    //System.out.println(e.getMessage());
    count=10;
}finally{
    transaction.commit();
}

~~~~~
if(session!=null){
    session.close();
}
~~~~~
return count;
}
}

```

Assign subject to teacher

UI

Enter Teacher Name: Vasudev

Enter subject Name: history-10

Assign subject to Teacher

[Go back to Admin Home page](#)

[Add Subject](#)

[Add Teacher](#)

[Assign more subjects to Teacher](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

On hitting the Assign subject to Teacher button

Enter Teacher Name:

Enter subject Name:

Assign subject to Teacher

[Go back to Admin Home page](#)

[Add Subject](#)

[Add Teacher](#)

[Assign more subjects to Teacher](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

1 subject assigned to teacher

Database:

1 • select * from teacher_subject

Teacher_teacherId	subjectList_subjectId
4	10
4	1
1	2
2	10
3	5
3	4
3	14
5	2
4	3
6	3
NULL	NULL

Code

```
public static int assignTeacherSubject(String teacherName,String subjectName){
    Session session=null;
    Transaction transaction=null;
    int count=0;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();
        //Subject s1=new Subject();
    }
```

```

/*Query query=session.createQuery("from Subject s where s.subjectName=:subjectName");
query.setParameter("subjectName", subjectName);
List subjectList=query.list();
System.out.println(subjectList);

```

```

Subject s1=new Subject();
if(subjectList.size()!=0){
    s1=(Subject)subjectList.get(0);

```

```

}
else{
    count=-1;

```

```

}
System.out.println(s1);*/
Teacher t1=new Teacher();

```

```

Query query1=session.createQuery("from Teacher t where t.teacherName=:teacherName");
query1.setParameter("teacherName", teacherName);
List teacherList=query1.list();
System.out.println(teacherList);

```

```

if(teacherList.size()==0){
    //t1.setTeacherName(teacherName);
    //session.save(t1);
    count=-2;

```

```

}else {
    t1=(Teacher)teacherList.get(0);
    Query query=session.createQuery("from Subject s where s.subjectName=:subjectName");
    query.setParameter("subjectName", subjectName);
    List subjectList=query.list();
    System.out.println(subjectList);

```

```

    Subject s1=new Subject();
    if(subjectList.size()!=0){
        s1=(Subject)subjectList.get(0);
        if(t1.getSubjectList().contains(s1)){
            count=0;
        }
        else{
            t1.getSubjectList().add(s1);
            session.merge(t1);
            count=1;
        }
    }

```

```

    }
    else{
        count=5;
        //session.flush();
    }

```

```

}catch(Exception e){
    System.out.println(e.getMessage());
}finally{

```

```

    transaction.commit();

```

```

    if(session!=null){
        session.close();
    }

```

```

return count;
}
}

```

Reports Generation

Students Report

On clicking the students Report link in Admin Home page

Students Report

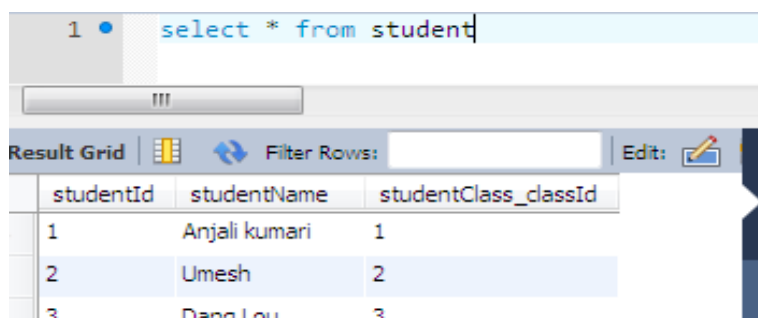
Student Id	Student Name	Student Class
1	Anjali kumari	class-1
2	Umesh	class-2
3	Dang Lou	class-3
4	Hitesh	class-4
5	Remo DeCosta	class-5
6	Sanjana	class-6
7	kishore	class-7
8	Stalin	class-8
9	K Sathish	class-9
10	Aswini Kumar	class-10
11	Satnam	class-7

[Go back to Admin Home page](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

Database:



The screenshot shows a database management interface. At the top, a SQL query is entered: `select * from student`. Below the query, there is a 'Result Grid' section. The grid has three columns: `studentId`, `studentName`, and `studentClass_classId`. The first three rows of data are visible:

studentId	studentName	studentClass_classId
1	Anjali kumari	1
2	Umesh	2
3	Dang Lou	3

4	Hitesh	4
5	Remo DeCosta	5
6	Sanjana	6
7	kishore	7
8	Stalin	8
9	K Sathish	9
10	Aswini Kumar	10
11	Satnam	7
NULL	NULL	NULL

E

Code:

```

public static List<Student> getStudentsList(){
    Session session=null;
    Transaction transaction=null;
    List studentList=null;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();

        Query query=session.createQuery("from Student");

        studentList=query.list();
        System.out.println(studentList);

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();

        if(session!=null){
            session.close();
        }
    }
}

```

Subjects Report

On clicking the Subjects Report link in Admin Home page

Subject Report

Subject Id	Subject Name	Subject Class
1	maths-10	class-10
2	physics-10	class-10
3	history-10	class-10
4	drawing-1	class-1
5	drawing-2	class-2
6	english-3	class-3
7	socialstudies-4	class-4
8	history-5	class-5
9	maths-6	class-6
10	history-6	class-6
11	physics-7	class-7
12	history-8	class-8
13	maths-9	class-9
14	chemistry-9	class-9
15	geography-10	class-10
16	english-5	class-5

[Go back to Admin Home page](#)

Database:

```
1 • select * from subject
```

subjectId	subjectName	subjectClass_classId
1	maths-10	10
2	physics-10	10
3	history-10	10
4	drawing-1	1
5	drawing-2	2
6	english-3	3
7	socialstudies-4	4
8	history-5	5
9	maths-6	6
10	history-6	6
11	physics-7	7
12	history-8	8
13	maths-9	9
14	chemistry-9	9

15	geography-10	10
16	english-5	5
★	NULL	NULL

Code:

```

public static List<Subject> getSubjectsList(){
    Session session=null;
    Transaction transaction=null;
    List subjectList=null;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();

        Query query=session.createQuery("from Subject");

        subjectList=query.list();
        System.out.println(subjectList);

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();

        if(session!=null){
            session.close();
        }
    }

    return subjectList;
}

```

Classes Report

On clicking the Classes Report link in Admin Home page

Classes Report

Class Id	Class Name
----------	------------

1	class-1
2	class-2
3	class-3
4	class-4
5	class-5
6	class-6
7	class-7
8	class-8
9	class-9
10	class-10
13	class-11

[Go back to Admin Home page](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

Code:

```

public static List<Class> getClassesList(){
    Session session=null;
    Transaction transaction=null;
    List classesList=null;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();

        Query query=session.createQuery("from Class");

        classesList=query.list();
        System.out.println(classesList);

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{

```

```

}
transaction.commit();
}
if(session!=null){
    session.close();
}
}

return classesList;
}
}

```

Teachers Report

On clicking the Teachers Report link in Admin Home page

Teachers Report

Teacher Id	Teacher Name	Teacher Class	Teacher Subject
1	Anand Kumar	[class-10]	[physics-10]
2	Renu Bhagat	[class-9, class-10, class-7]	[history-6]
3	Daya Kumar	[class-5, class-1]	[drawing-2, drawing-1, chemistry-9]
4	Sanjeev Reddy	[class-2, class-7]	[history-6, maths-10, history-10]
5	Irfan Khan	[class-9, class-6]	[physics-10]
6	Vasudev	[class-11]	[history-10]

[Go back to Admin Home page](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

Database:

The screenshot shows a database management interface. At the top, a SQL query is entered: `select * from teacher`. Below the query, there is a 'Result Grid' section. It includes a 'Filter Rows' input field and an 'Edit' button. The result grid displays the following data:

teacherId	teacherName
1	Anand Kumar
2	Renu Bhagat
3	Daya Kumar
4	Sanjeev Reddy
5	Irfan Khan
6	Vasudev

0	vasudev
NULL	NULL

Code:

```

public static List<Teacher> getTeachersList(){
    Session session=null;
    Transaction transaction=null;
    List teacherList=null;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();

        Query query=session.createQuery("from Teacher");

        teacherList=query.list();
        System.out.println(teacherList);

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();

        if(session!=null){
            session.close();
        }
    }
    return teacherList;
}
}

```

Master Report

On clicking the Master Report link in Admin Home page

Class Master Report

Enter Class Name:

Teachers associated with
Subjects associated with
Students associated with

[Go back to Admin Home page](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

On hitting the Get Master Report button

Class Master Report

Enter Class Name:

Get Master Report

Teachers associated with class-10
Renu Bhagat
Anand Kumar
Subjects associated with class-10
maths-10
physics-10
history-10
geography-10
Students associated with class-10
Aswini Kumar

[Go back to Admin Home page](#)

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

Code:

```
public static List<Teacher> getMasterReportTeacher(String className){
    Session session=null;
    Transaction transaction=null;
    List teacherNames=null;
```

```

try{
    Configuration configuration=new Configuration().configure();
    configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
    configuration.addAnnotatedClass(learnersacademy.models.Student.class);
    configuration.addAnnotatedClass(learnersacademy.models.Class.class);
    configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
    configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

    StandardServiceRegistryBuilder builder=new
        StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
    SessionFactory factory=configuration.buildSessionFactory(builder.build());
    session=factory.openSession();
    transaction=session.beginTransaction();

    String sqlQuery="select t.teacherName from teacher t,teacher_class tc"+
        "where t.teacherId=tc.Teacher_teacherId"+
        " and tc.classList_classId =:className";
    String hqlQuery="select t.teacherName from Teacher t inner join t.classList tc "+
        "where tc.className=:className";

    /*Query query=session.createQuery(sqlQuery)
        .addEntity("t",Teacher.class)
        ;

    query.setInteger("className",className);*/
    Query query=session.createQuery(hqlQuery);
    query.setParameter("className", className);
    teacherNames=query.list();

```

```

        System.out.println(teacherNames);

```

```

}catch(Exception e){
    System.out.println(e.getMessage());
}finally{
    transaction.commit();
    ~~~~~
    if(session!=null){
        session.close();
    }
    ~~~~~
    return teacherNames;
}
}

```

```

② public static List<Subject> getMasterReportSubject(String className){
    Session session=null;
    Transaction transaction=null;
    List subjectNames=null;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

```

```

StandardServiceRegistryBuilder builder=new
    StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
SessionFactory factory=configuration.buildSessionFactory(builder.build());
session=factory.openSession();
transaction=session.beginTransaction();

Query query=session.createQuery("select s.subjectName from Subject s where s.subjectClass.className="
query.setParameter("className", className);
subjectNames=query.list();
System.out.println(subjectNames);

```

```

}catch(Exception e){
    System.out.println(e.getMessage());
}finally{
    transaction.commit();
    if(session!=null){
        session.close();
    }
}

```

```

return subjectNames;
}
}

```

```

public static List<Subject> getMasterReportStudent(String className){
    Session session=null;
    Transaction transaction=null;
    List studentNames=null;

    try{
        Configuration configuration=new Configuration().configure();
        configuration.addAnnotatedClass(learnersacademy.models.Admin.class);
        configuration.addAnnotatedClass(learnersacademy.models.Student.class);
        configuration.addAnnotatedClass(learnersacademy.models.Class.class);
        configuration.addAnnotatedClass(learnersacademy.models.Subject.class);
        configuration.addAnnotatedClass(learnersacademy.models.Teacher.class);

        StandardServiceRegistryBuilder builder=new
            StandardServiceRegistryBuilder().applySettings(configuration.getProperties());
        SessionFactory factory=configuration.buildSessionFactory(builder.build());
        session=factory.openSession();
        transaction=session.beginTransaction();

        Query query=session.createQuery("select st.studentName from Student st where st.studentClass.classN
        query.setParameter("className", className);
        studentNames=query.list();
        System.out.println(studentNames);

    }catch(Exception e){
        System.out.println(e.getMessage());
    }finally{
        transaction.commit();
    }
}

```



```
~~~~~
    if(session!=null){
~~~~~
        session.close();
~~~~~
    }

~~~~~
return studentNames;
}

}
```

Logout Functionality

UI for logout functionality

Admin Utility

[Add Subjects for a given class](#)
[Add Students for a given class](#)
[Add Teachers](#)
[Add Classes](#)
[Assign class to a given Teacher](#)
[Assign subjects to a given teacher](#)

Admin Reports

[Students Report](#)
[Subjects Report](#)
[Teachers Report](#)
[Classes Report](#)
[Master Report](#)

[Log Out](#)

©Copyright 2021 Learner's Academy

On clicking the Log out link

Admin Name:

Password :

Login

©Copyright 2021 Learner's Academy

This application has been designed and developed by Priyadarshi Panigrahi Email-priyadarshi_05@yahoo.co.in

Disabling the back button



Webpage has expired

Most likely cause:

- The local copy of this webpage is out of date, and the website requires that you download it again.

Something to try:

- Click on the Refresh button on the toolbar to reload the page. After refreshing, you might need to navigate to the specific webpage again, or re-enter information.

More information

Code

```
<body>
<%
response.setHeader("Cache-Control", "no-cache,no-store,must-revalidate");
//response.setHeader("Cache-Control", "no-store");
response.setHeader("Pragma", "no-cache");
response.setDateHeader("Expires", 0);
%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
Logged in Successfully at <%= new Date() %>
<br/>
<br/>
<h1 style="color:blue">Admin Utility</h1>
<br/>
<br/>
<a href="AddSubject.jsp">Add Subjects for a given class</a><br/>
<a href="AddStudent.jsp">Add Students for a given class</a><br/>
<a href="AddTeacher.jsp">Add Teachers</a><br/>
<a href="AddClasses.jsp">Add Classes</a><br/>
<a href="AssignTeacherClass.jsp">Assign class to a given Teacher</a><br/>
<a href="AssignTeacherSubject.jsp">Assign subjects to a given teacher</a><br/>

<br/>
<br/>
<h1 style="color:blue">Admin Reports</h1>
<br/>
<br/>
<a href="StudentsList">Students Report</a><br/>
<a href="SubjectList">Subjects Report</a><br/>

<a href="TeacherList">Teachers Report</a><br/>
<a href="ClassesList">Classes Report</a><br/>
<a href="MasterReport.jsp">Master Report</a><br/>

<br/>
```



xception {







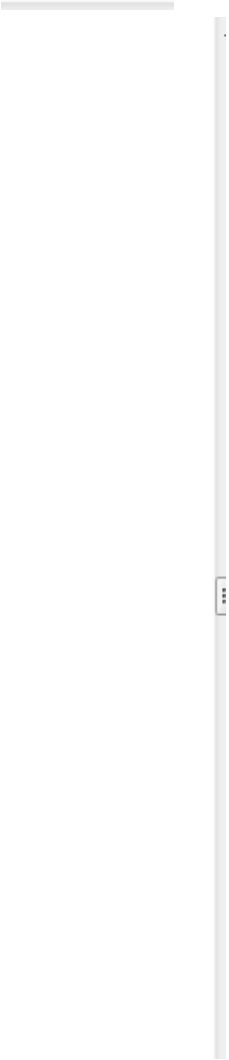














```
e=:className");
```

```
lame=:className");
```





