Project Name	Back-end Admin for Learner's Academy
Developers Name	Prathamesh Chinchamalatpure
Phase	Become a back-end expert
Part	Writeup

This document contains the following contents:

1. Sprint Planned and task achieved in them

2. Algorithm and flowcharts of the application

3. Core concepts used in the project

4. Links to the GitHub repository

5. Unique Selling Points

6. Conclusions

This project is hosted at https://github.com/Prathameshcgitnew/Phase3ProjectDemo.git

The project is developed by Prathamesh Chinchamalatpure

1.Sprint Planning and Task Achieved in them

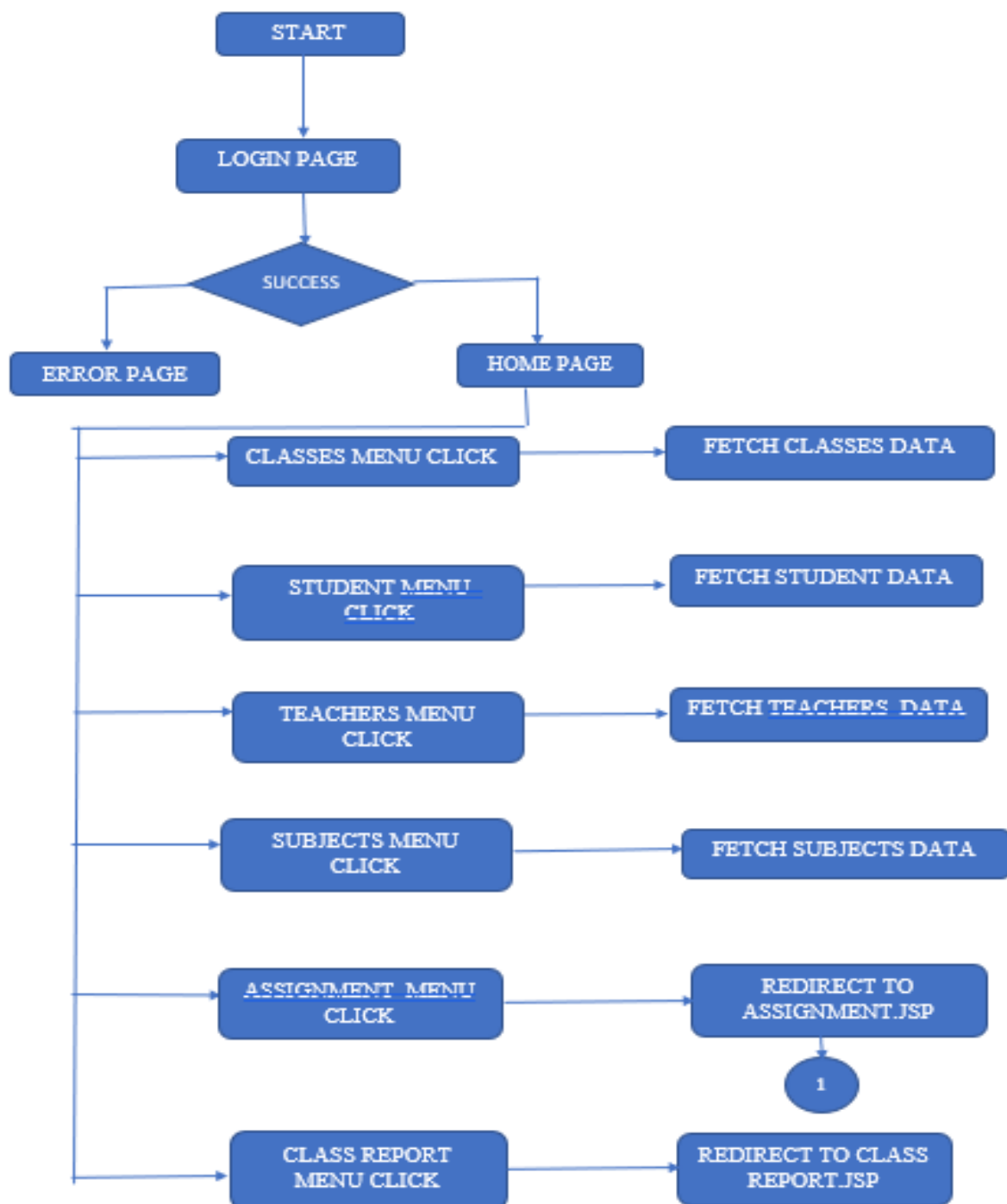This project is planned to be completed in 2 sprint. Task that are assumed to be completed are:

 1   Developing flow for project

 2   Initializing git repositories to track changes as project progresses

 3   Writing Java code to fulfill requirement for the project

 4   Testing project by giving diverse input

 5   Pushing code to github

 6   Creating document of specification highlighting application appearances , capabilities
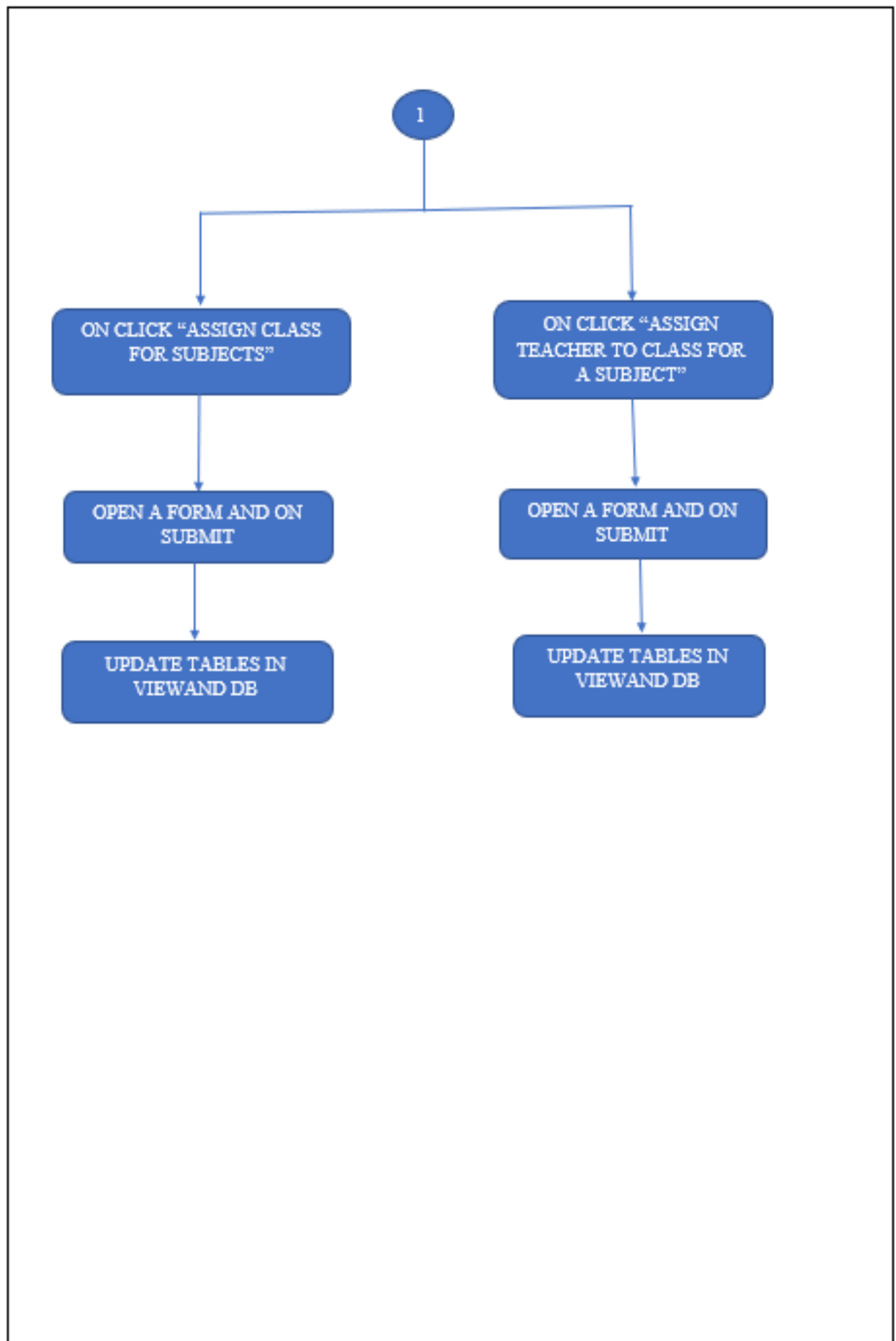
   and user interactions

2. Core concepts required in project

String, Collection Framework , Control constructs, Exception Handling ,Servlet ,JSP,Hibernate,Filter
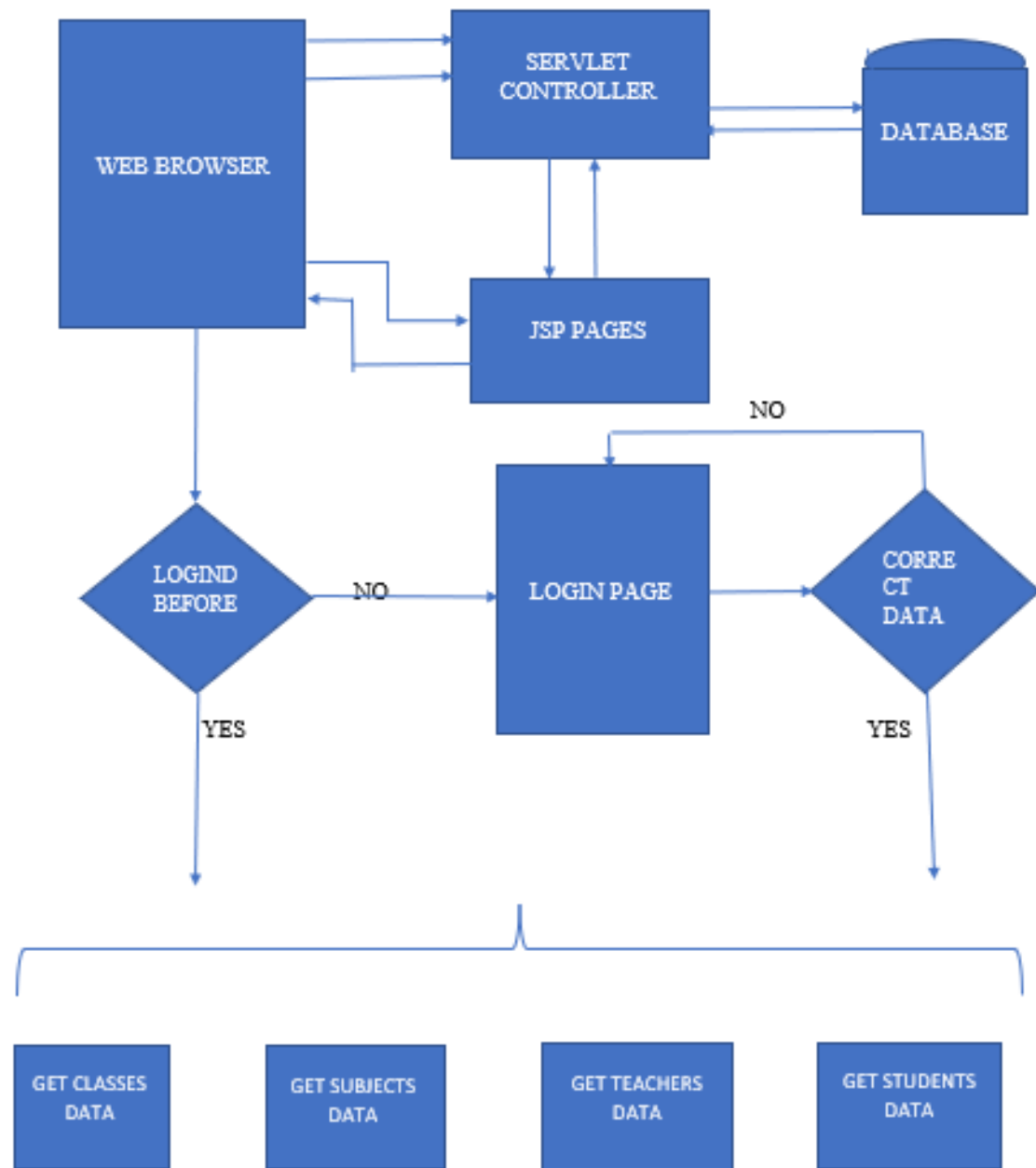
3.Flow for project

FUNCTIONAL FLOWCHART :

```
                    START
                      |
                LOGIN PAGE
                      |
                  SUCCESS
          /                      \
   ERROR PAGE                 HOME PAGE
                                  |
   CLASSES MENU CLICK ──────► FETCH CLASSES DATA

   STUDENT MENU CLICK ──────► FETCH STUDENT DATA

   TEACHERS MENU CLICK ─────► FETCH TEACHERS DATA

   SUBJECTS MENU CLICK ─────► FETCH SUBJECTS DATA

   ASSIGNMENT MENU CLICK ───► REDIRECT TO ASSIGNMENT.JSP
                                       |
                                       1

   CLASS REPORT MENU CLICK ─► REDIRECT TO CLASS REPORT.JSP
```

```
                                    ┌───┐
                                    │ 1 │
                                    └─┬─┘
                      ┌───────────────┴───────────────┐
                      ▼                                ▼
         ┌──────────────────────┐        ┌──────────────────────┐
         │  ON CLICK "ASSIGN     │        │   ON CLICK "ASSIGN    │
         │   CLASS FOR SUBJECTS" │        │  TEACHER TO CLASS FOR │
         │                       │        │      A SUBJECT"       │
         └───────────┬───────────┘        └───────────┬───────────┘
                     ▼                                ▼
         ┌──────────────────────┐        ┌──────────────────────┐
         │  OPEN A FORM AND ON   │        │  OPEN A FORM AND ON   │
         │       SUBMIT          │        │       SUBMIT          │
         └───────────┬───────────┘        └───────────┬───────────┘
                     ▼                                ▼
         ┌──────────────────────┐        ┌──────────────────────┐
         │   UPDATE TABLES IN    │        │   UPDATE TABLES IN    │
         │    VIEWAND DB         │        │    VIEWAND DB         │
         └──────────────────────┘        └──────────────────────┘
```

STRUCTURAL FLOWCHART:

WEB BROWSER

SERVLET CONTROLLER

DATABASE

JSP PAGES

LOGIND BEFORE

NO

LOGIN PAGE

NO

CORRECT DATA

YES

YES

GET CLASSES DATA

GET SUBJECTS DATA

GET TEACHERS DATA

GET STUDENTS DATA

4 . Product capabilities, appearance and user interactions

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

1 Creating the project in Eclipse

2 Writing a program in Java for the entry point of the application (Login.jsp & Login.java)

3 Writing a program in jsp page to display options available for the Admin (Navigation bar for each JSP page)

4 Writing a program in Java to filter the requests (FilterServletRequests.java)

5 Writing a program in Java to perform database connectivity. (HibernateUtil.java)

6 Writing a programs for functional flows (Different Servlet Programs)

7 Writing a program to for entities required for database.

8 Pushing the code to the Git repository.

1 Creating the project in Eclipse

1. Open Eclipse.
2. File->New->Project->Maven->Maven Project
3. Maven Project-> select archetype->Enter GroupId , ArtifactId
4. Click Finish.
5. Create package and create classes, servlets, jsp pages,filter.

2 Writing a program in Java for the entry point of the application (Login.jsp & Login.java)

1.Creating and developing Login.jsp it will show login as shown in image.

2.Writing a program for Login using admin credential.

Login.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>


<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login</title>
 <link type="text/css" rel="stylesheet" href="css/login.css">
</head>
<body>

    <div class="center">
    <div class="center1">
```

```html
<h1> Admin Login </h1>

 <form action="Login" method="post">

      <label>Username  : </label>
       <input type="text" placeholder="Enter Username" name="username"
required>
       <br><br>
       <label>Password  : </label>
       <input type="password" placeholder="Enter Password" name="password"
required>
       <br><br>
       <div class="center1">
       <button type="submit">Login</button>
       </div>
       <br/>

       <br>
       <input type="checkbox" checked="checked"> Remember me

     </form>
    </div>
</div>

</body>
</html>
```
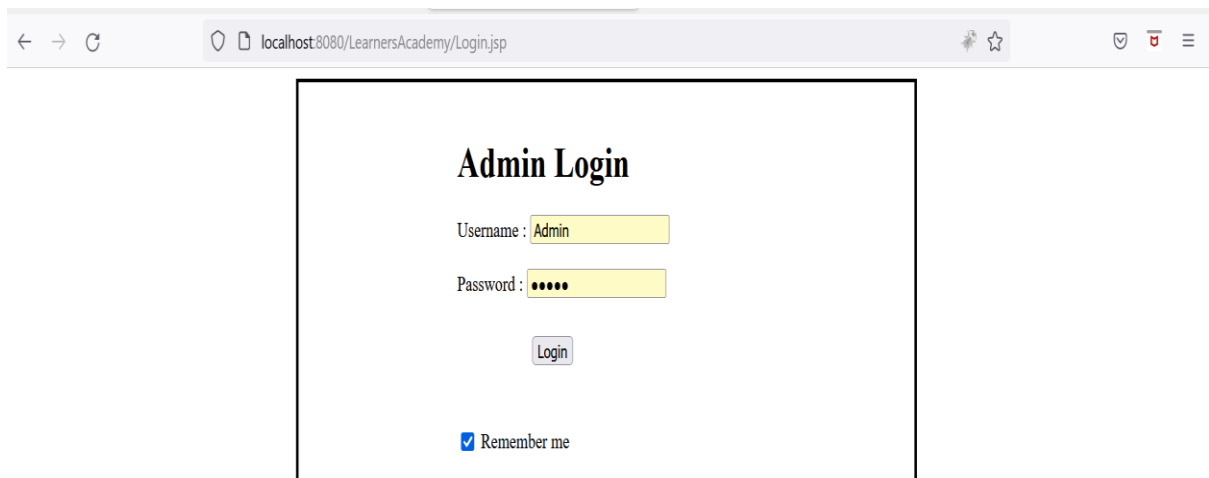
Admin Login

Username : Admin

Password : •••••

Login

☑ Remember me

Login.java

```java
package com.servlets;

import java.io.IOException;
import java.io.PrintWriter;

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;

/**
 * Servlet implementation class Login
 */
@WebServlet("/com.servlets.Login")
public class Login extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#HttpServlet()
	 */
	public Login() {
		super();
		// TODO Auto-generated constructor stub
	}

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		// TODO Auto-generated method stub

		Cookie userName=new
Cookie("userName",request.getParameter("username"));
		Cookie passWord=new
Cookie("passWord",request.getParameter("password"));

		Cookie cookie=null;
		Cookie []cookies =request.getCookies();
		Boolean check=false,check1=false;
		PrintWriter out=response.getWriter();
		Boolean cookieUserNameExists=false,cookiePasswordExists=false;
		//response.sendRedirect("Home.jsp");

		for(Cookie c:cookies) {
			if(c.getName().equals("userName")) {
				cookieUserNameExists=true;
			}
			if(c.getName().equals("passWord")) {
				cookiePasswordExists=true;
			}
		}
		if(!cookieUserNameExists && !cookiePasswordExists)
```

```
                {
                        if((userName.getValue().equals("Admin"))  &&
(passWord.getValue().equals("Admin")) )
                        {
                                response.addCookie(userName);
                                response.addCookie(passWord);
                                userName.setMaxAge(3600);
                                passWord.setMaxAge(3600);
                                RequestDispatcher
rd=request.getRequestDispatcher("/Home.jsp");
                                rd.forward(request, response);
                        }else {
                                out.println("Enter Valid credential");
                        }
                }else {
                        if((userName.getValue().equals("Admin"))  &&
(passWord.getValue().equals("Admin")) )
                        {
                                RequestDispatcher
rd=request.getRequestDispatcher("/Home.jsp");
                                rd.forward(request, response);
                        }else {
                                out.println("Enter Valid credential");
                        }

                }

        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                doGet(request,response)


        }

}
```

3 Writing a program in jsp page to display options available for the Admin (Navigation bar for each JSP page)

1. Creating and developing jsp page diplaying navigation sidebar.

2. Page is shown in image.

Home.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
```

```html
<title></title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/dashboard.css">
<body>

<!-- Sidebar -->
<div class="w3-sidebar w3-light-grey w3-bar-block" style="width:25%">
  <h3 class="w3-bar-item">ADMIN PORTAL</h3>
  <a href="class?command=classesfetch" class="w3-bar-item w3-button">CLASSES</a>
  <a href="teachers?command=teachersfetch" class="w3-bar-item w3-button">TEACHERS</a>
  <a href="students?command=studentsfetch" class="w3-bar-item w3-button">STUDENTS</a>
  <a href="subjects?command=subjectsfetch" class="w3-bar-item w3-button">SUBJECTS</a>
  <a href="Assignment.jsp" class="w3-bar-item w3-button">ASSIGNMENT</a>
   <a href="reporting?command=reportcreation" class="w3-bar-item w3-button">CLASS-REPORT</a>
     <a href="Login.jsp" class="w3-bar-item w3-button">LOGOUT</a>

</div>

<div style="margin-left:25%">
<div class="w3-container w3-grey">
  <div style="text-align:center">
     <h1>Admin Portal</h1>
  </div>
</div>
<div class="w3-container">
Please use navigation sidebar to navigate.
</div>
</div>
 </body>
 </html>
```
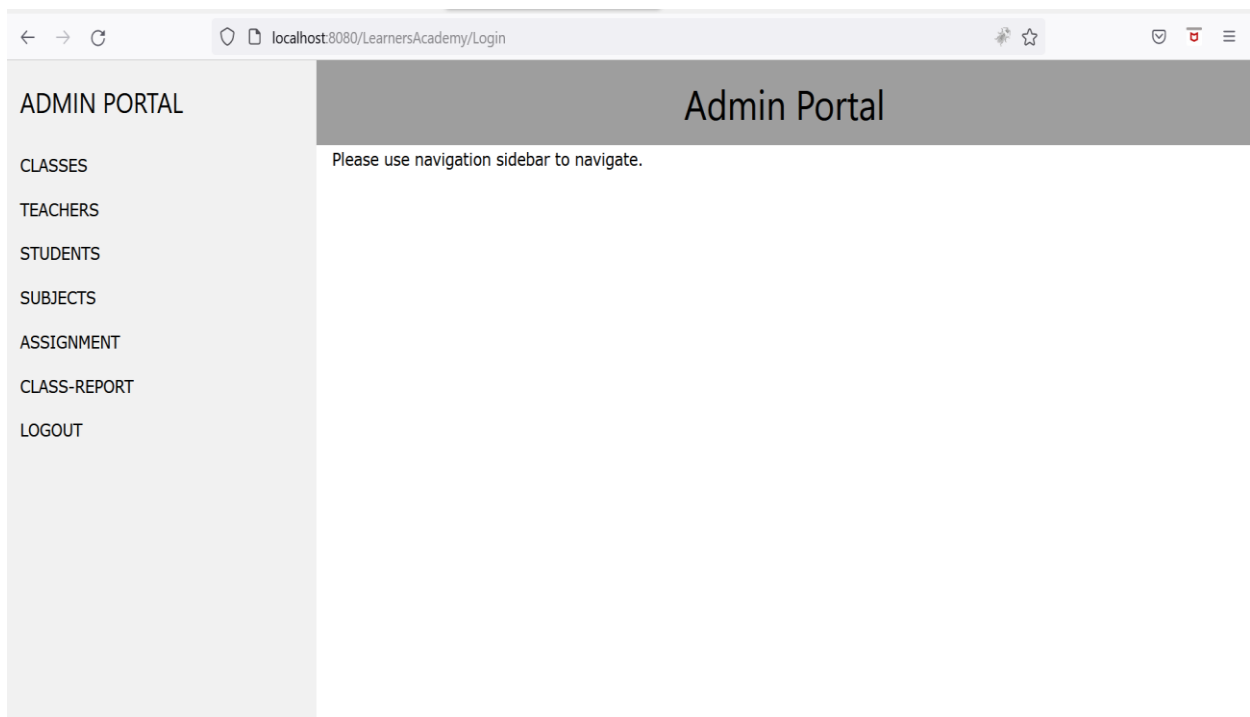**Image:**

4 Writing a program in Java to filter the requests (FilterServletRequests.java)

FilterServletRequests.java

```java
package com.filters;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServletRequest;

/**
 * Servlet Filter implementation class FilterServletRequests
 */
public class FilterServletRequests implements Filter {

    /**
     * Default constructor.
     */
    public FilterServletRequests() {
        // TODO Auto-generated constructor stub
    }

    /**
     * @see Filter#destroy()
     */
    public void destroy() {
        // TODO Auto-generated method stub
    }
```

```java
/**
 * @see Filter#doFilter(ServletRequest, ServletResponse, FilterChain)
 */
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
    // TODO Auto-generated method stub
    // place your code here
    HttpServletRequest req=(HttpServletRequest)request;
    Cookie []cookies =req.getCookies();
    Boolean check=false,check1=false;
    PrintWriter out=response.getWriter();
    Boolean cookieUserNameExists=false,cookiePasswordExists=false;
    //response.sendRedirect("Home.jsp");
    if(cookies==null) {
        out.println("Kindly  Login . You are redirecting to Login Page");
        RequestDispatcher rd=request.getRequestDispatcher("Login.jsp");
        rd.forward(request,response);
    }else {
        for(Cookie c:cookies) {
            if(c.getName().equals("userName")) {
                cookieUserNameExists=true;
            }
            if(c.getName().equals("passWord")) {
                cookiePasswordExists=true;
            }
        }
        if(cookieUserNameExists && cookiePasswordExists) {
            // pass the request along the filter chain
            chain.doFilter(request, response);
        }else {
            out.println("You are not Login. Please Login.");
        }
    }
}
```

```
                }

        }

//         chain.doFilter(request, response);

}

/**

 * @see Filter#init(FilterConfig)

 */

public void init(FilterConfig fConfig) throws ServletException {

        // TODO Auto-generated method stub

}

}
```

5 Writing a program in Java to perform database connectivity. (HibernateUtil.java)

HibernateUtil.java

```java
package com.utilities;

import org.hibernate.SessionFactory;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class HibernateUtil {

        private static final SessionFactory sessionFactory;


        static {
                try{
                        StandardServiceRegistry serviceregistry=new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
                        Metadata metadata=new
MetadataSources(serviceregistry).getMetadataBuilder().build();
                        sessionFactory=metadata.getSessionFactoryBuilder().build();
                }catch(Throwable th) {
                        throw new ExceptionInInitializerError();
                }
        }

        public static SessionFactory getSessionFactory() {
                return sessionFactory;
```

```
            }

}
```

6 Writing a programs for functional flows (Different Servlet Programs)

1.ClassesRetrieval.java

package com.servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.RequestDispatcher;
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 org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.entities.Class;
import com.utilities.HibernateUtil;

```
/**
 * Servlet implementation class ClassesRetrieval
 */
@WebServlet("/com.servlets.ClassesRetrieval")
public class ClassesRetrieval extends HttpServlet {
        private static final long serialVersionUID = 1L;

   /**
    * @see HttpServlet#HttpServlet()
    */
   public ClassesRetrieval() {
     super();
     // TODO Auto-generated constructor stub
   }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // TODO Auto-generated method stub
```

```java
                //response.getWriter().append("Served at:
").append(request.getContextPath());
                PrintWriter out=response.getWriter();


                SessionFactory factory=HibernateUtil.getSessionFactory();

                Session session=factory.openSession();

    List<Class> list = session.createQuery("From Class").list();

                session.close();
                //request.setAttribute("Product_LIST", students);

                /*for(EProduct p: list)
    {
        out.println("ID:" + String.valueOf(p.getID()) + ",Name: "+p.getName() + ",Price:
"+String.valueOf(p.getPrice() + ",Date Added: "+p.getDateAdded().toString() + "<br>"));
    }*/

                request.setAttribute("class", list);

                RequestDispatcher rd=request.getRequestDispatcher("Classes.jsp");
                rd.forward(request, response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // TODO Auto-generated method stub
                //doGet(request, response);
        }

}
```

2.ClassReport.java

```java
package com.servlets;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;
```

```java
import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import com.entities.Class;

import com.entities.Students;

import com.utilities.HibernateUtil;
/**
 * Servlet implementation class ClassReport
 */
public class ClassReport extends HttpServlet {
        private static final long serialVersionUID = 1L;
   /**
    * @see HttpServlet#HttpServlet()
    */
   public ClassReport() {
     super();
     // TODO Auto-generated constructor stub
   }
        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // TODO Auto-generated method stub
                // Section
                //Teachers Name from class Table as per section
                //Subjects for that section from class table
                //Students from student table as per section
```

```java
			SessionFactory factory=HibernateUtil.getSessionFactory();

			Session session=factory.openSession();

	List<Class> list = session.createQuery("From Class").list();

			session.close();

			ClassReport classReport=new ClassReport();

			List<Students> studentsListOfSection1=classReport.getStudentsBySection(1);

			List<Students> studentsListOfSection2=classReport.getStudentsBySection(2);


			request.setAttribute("class", list);

			request.setAttribute("studentsOfSection1", studentsListOfSection1);

			request.setAttribute("studentsOfSection2", studentsListOfSection2);

			RequestDispatcher rd=request.getRequestDispatcher("ClassReport.jsp");

			rd.forward(request, response);

		}


	/**

	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)

	 */

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

			// TODO Auto-generated method stub

			doGet(request, response);

		}



	public List<Students> getStudentsBySection(int section){


			SessionFactory factory=HibernateUtil.getSessionFactory();
```

```java
                Session session=factory.openSession();

                List<Students> studentsList=session.createQuery("From Students").list();

                session.close();

                List<Students> listOfSection=new ArrayList<Students>();

                for(Students s:studentsList) {
                        if(s.getSection()==section) {
                                listOfSection.add(s);
                        }
                }
                return listOfSection;
        }
}
```

3.StudentsRetrieval

```java
package com.servlets;

import java.io.IOException;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.entities.Students;
import com.utilities.HibernateUtil;

/**
 * Servlet implementation class StudentsRetrieval
 */
```

```java
public class StudentsRetrieval extends HttpServlet {
        private static final long serialVersionUID = 1L;

   /**
    * @see HttpServlet#HttpServlet()
    */
   public StudentsRetrieval() {
      super();
      // TODO Auto-generated constructor stub
   }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // TODO Auto-generated method stub
                SessionFactory factory=HibernateUtil.getSessionFactory();

                Session session=factory.openSession();

                List<Students> list=session.createQuery("From Students").list();

                session.close();

                request.setAttribute("students", list);

                RequestDispatcher rd=request.getRequestDispatcher("Students.jsp");
                rd.forward(request, response);


                //response.getWriter().append("Served at :").append("StudentsRetrieval");
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}
```

5.SubjectsRetrieval.java

package com.servlets;

```java
import java.io.IOException;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.entities.Subjects;
import com.utilities.HibernateUtil;

/**
 * Servlet implementation class SubjectsRetrieval
 */
public class SubjectsRetrieval extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SubjectsRetrieval() {
        super();
        // TODO Auto-generated constructor stub
    }
```

```java
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub

        SessionFactory factory=HibernateUtil.getSessionFactory();

        Session session=factory.openSession();

        List<Subjects> list=session.createQuery("From Subjects").list();

        session.close();

        request.setAttribute("subjects", list);

        RequestDispatcher rd=request.getRequestDispatcher("Subjects.jsp");
        rd.forward(request, response);
        //response.getWriter().append("Served at: ").append("SubjectsRetrieval");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
```

}

6.TeachersRetrieval

package com.servlets;

import java.io.IOException;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.entities.Teachers;
import com.utilities.HibernateUtil;

```
/**
 * Servlet implementation class TeachersRetrieval
 */
public class TeachersRetrieval extends HttpServlet {
        private static final long serialVersionUID = 1L;

  /**
   * @see HttpServlet#HttpServlet()
   */
  public TeachersRetrieval() {
```

```java
        super();
        // TODO Auto-generated constructor stub
    }


    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            // TODO Auto-generated method stub
            SessionFactory factory=HibernateUtil.getSessionFactory();

            Session session=factory.openSession();

            List<Teachers> list=session.createQuery("From Teachers").list();

            session.close();

            request.setAttribute("teachers", list);

            RequestDispatcher rd=request.getRequestDispatcher("Teachers.jsp");
            rd.forward(request, response);

            //response.getWriter().append("Served at: ").append("TeachersRetrieval");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
```

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

                // TODO Auto-generated method stub

                doGet(request, response);

        }


}
```

7.UpdateClasses.java

```java
package com.servlets;


import java.io.IOException;

import java.io.PrintWriter;


import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;


import com.entities.*;

import com.entities.Class;

import com.utilities.HibernateUtil;
/**
 * Servlet implementation class UpdateClasses
 */
public class UpdateClasses extends HttpServlet {

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

```java
/**
 * @see HttpServlet#HttpServlet()
 */
public UpdateClasses() {
    super();
    // TODO Auto-generated constructor stub
}


    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            // TODO Auto-generated method stub
            response.getWriter().append("Served at: ").append(request.getContextPath());
    }


    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            // TODO Auto-generated method stub
        Class c=new Class();//form com.entities.Class


            String section=request.getParameter("sections");
            String teacher=request.getParameter("teachers");
            String subject=request.getParameter("subjects");
            String time =request.getParameter("time");
            c.setSection(Integer.parseInt(section));
```

```java
            c.setSubject(subject);

            c.setTeacher(teacher);

            c.setTime(time);

            PrintWriter out=response.getWriter();

            Transaction transaction=null;

            //out.println(section+" "+teacher+" "+subject+" "+time);

            try {

            SessionFactory factory=HibernateUtil.getSessionFactory();

            //doGet(request, response);


            Session session=factory.openSession();


        transaction=session.beginTransaction();


            session.save(c);


            transaction.commit();

            }catch(Exception e) {

                    if(transaction!=null) {

                            transaction.rollback();

                    }


                    e.printStackTrace();

            }


    }


}
```

7 .Writing a program to for entities required for database.

1.Class.java

```java
package com.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="class")
public class Class {

        @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
        @Column(name="ID")
        private int id;

        @Column(name="section")
        private int section;

        @Column(name="teacher")
        private String teacher;

        @Column(name="subject")
        private String subject;

        @Column(name="time")
        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 Class() {}
        @Override
        public String toString() {
                return "Class [id=" + id + ", section=" + section + ", teacher=" + teacher + ",
subject=" + subject + ", time="
                                        + time + "]";
        }
```

```java
        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;
        }


}
```

2.Students.java

```java
package com.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="students")
public class Students {
```

```java
@Id @GeneratedValue
@Column(name="Id")
private int id;

@Column(name="fname")
private String fname;

@Column(name="lname")
private String lname;

@Column(name="age")
private int age;


@Column(name="section")
private int section;

//@Column(name="assignedclass")
//private int aclass;

public Students(int id, String fname, String lname, int age, int aclass) {
        super();
        this.id = id;
        this.fname = fname;
        this.lname = lname;
        this.age = age;
        //this.aclass = aclass;
}



public int getSection() {
        return section;
}



public void setSection(int section) {
        this.section = section;
}



public Students() {

}



public int getId() {
```

```java
                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 aclass;
        }

        public void setAclass(int aclass) {
                this.aclass = aclass;
        }
        */

}
```

3.Subjects.java

```java
package com.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
```

```java
@Entity
@Table(name="subjects")
public class Subjects {

        @Id @GeneratedValue
        @Column(name="ID")
        private int id;

        @Column(name="subjectname")
        private String name;

        @Column(name="shortcut")
        private String shortcut;


        public Subjects(int id, String name, String shortcut) {
                super();
                this.id = id;
                this.name = name;
                this.shortcut = shortcut;
        }


        public Subjects() {

        }


        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getShortcut() {
                return shortcut;
        }
        public void setShortcut(String shortcut) {
                this.shortcut = shortcut;
        }


}
```

4.Teachers.java

package com.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

```java
import javax.persistence.Table;

@Entity
@Table(name="teachers")
public class Teachers {

        @Id @GeneratedValue
        @Column(name="id")
        private int id;

        @Column(name="fname")
        private String fname;

        @Column(name="lname")
        private String lname;

        @Column(name="age")
        private int age;

        public Teachers() {

        }

        public Teachers(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;
        }

}
```

8.Pushing the code to GitHub repository

1. Open your command prompt and navigate to the folder where you have created your files.

2. Initialize repository using the following command: **git init**

3. Add all the files to your git repository using the following command: **git add .**

4. Commit the changes using the following command: **git commit . -m**

5. Push the files to the folder you initially created using the following command:

    git push -u origin master

Unique Selling Points of the Application

1. Application is single user application. So only admin can access it with right credentials.

2. Application is robust for unauthorized access . The admin need to login first then only admin can access its internal information.

3. Application have given ultimate feature to assign teachers, time, subject .

4. Its dynamic nature to change class-report and classes information as per admin assignment makes it unique.

5. User Interaction is smooth as all navigational options are given on dashboard.

Conclusion

Application enhancement is possible at following point:

1. Allowing admin to add any number of students to the list

2. Allowing admin to assign any number of classes

3. Allowing admin retrieval of time when last time application was accessed