DEVELOPMENT OF ANTI-RAGGING PORTAL

Project submitted to

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

In partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATIONS, 2018

BY

Paroj Chakraborty

Examination Roll: MCA186011

Registration No: 133673 of 2015-2016

Under the guidance of

Prof. Chandan Mazumdar

Professor, Department of Computer Science Engineering

Jadavpur University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

TO WHOM IT MAY CONCERN

I hereby recommend that the project entitled "Development of Anti-Ragging Portal" prepared under my supervision and guidance at Jadavpur University, Kolkata by PAROJ CHAKRABORTY *(Reg. No. 133673 of 2015 – 16, Class Roll No. 00151050300 of 2015-16), may be accepted in partial fulfillment for the degree of Master of Computer Applications in the Faculty of Engineering and Technology, Jadavpur University, during the academic year 2017 – 2018. I wish him every success in life.*

……..
Prof. (Dr.) Ujjwal Maulik
Head of the Department
Department of Computer Science and Engineering
Jadavpur University, Kolkata – 700032.

……
Prof. Chandan Mazumdar
Project Supervisor,
Department of Computer Science and Engineering
Jadavpur University, Kolkata – 700032.

……..
Prof. (Dr.) Chiranjib Bhattacharjee
Dean, Faculty council of Engg. & Tech.
Jadavpur University, Kolkata – 700032.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC PROJECT

I hereby declare that this project contains literature survey and original research work by the undersigned candidate, as part of his MASTER OF COMPUTER APPLICATIONS studies. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material results that are not original to this work.

NAME: PAROJ CHAKRABORTY

ROLL NUMBER: 001510503011

PROJECT TITLE: Development of Anti-Ragging Portal

SIGNATURE WITH DATE:

JADAVPUR UNIVERSITY
FACULTY OF ENGINEERING AND TECHNOLOGY

CERTIFICATE OF APPROVAL

The forgoing project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR
EVALUATION OF PROJECT:

1. _____
(Signature of the Supervisor)

2._____
(Signature of the Examiner)

ACKNOWLEDGEMENT

I express my honest and sincere thanks and humble gratitude to my respected teacher and guide *Prof. Chandan Mazumdar*, Professor of the Department of Computer Science & Engineering, Jadavpur University, for his exclusive guidance and entire support in completing and producing this project successfully. I am very much indebted to him for the constant encouragement, and continuous inspiration that he has given to me. The above words are only a token of my deep respect towards him for all he has done to take my project to the present shape.

I would like to thank *Mr. Preetam Mukherjee* and *Mr. Projjal Saha* for valuable support and suggestions to the activities of the project.

Finally, I convey my real sense of gratitude and thankfulness to my family members, specially my elder sister, for being an endless source of optimism and positive thoughts; and last but not the least, my father & mother for their unconditional support, without which I would hardly be capable of producing this huge work.

PAROJ CHAKRABORTY
Examination Roll: MCA186011
Registration No: 133673 of 2015 – 2016

Contents

CHAPTER – 1

INTRODUCTION

1. Introduction:

The struts framework is an open source framework for creating well-structured web-based applications. It is the product of the Apache software foundation which is basically used for developing web applications in Java. The struts framework is based on the Model View Controller (MVC) paradigm which distinctly separates all the three layers - Model (state of the application), View (presentation) and Controller (controlling the application flow). This makes struts different from conventional JSP applications where sometimes logic, flow and UI are mingled in a Java Server Page.

The struts framework is a complete web framework as it provides complete web form components, validators, error handling, internationalization, tiles and more. Struts framework provides its own Controller component. It integrates with other technologies for both Model and View components. Struts can integrate well with Java Server Pages (JSP), Java Server Faces (JSF), JSTL, Velocity templates and many other presentation technologies for View. For Model, Struts works great with data access technologies like JDBC, Hibernate, EJB and many more.

When the framework's controller receives a request, it uses the configuration file, struts-config.xml to find out the correct routing information. Based on the information configured in the configuration file, it internally invokes an Action class. The Action class interacts with the Model (also called the business layer) to access or update the underlying data in database/ file. The framework includes ActionForm classes to transfer data between Model and View.

CHAPTER – 2

PROBLEM DESCRIPTION & SRS

2.1 Problem Description:

An anti-ragging portal is to be developed as a web application in Struts 1.3 web framework which involves the whole cycle of registration and complaint by students, enquiry regarding the complaints, actions and decision taken after enquiry and result shown to the particular student.

There are two major processes involved:

A. **Student Part:**

- Firstly, a student has to register to the student portal.
- Then he/she can complaint after logging in to the portal.
- Once the complaint is lodged, the student can open his/her complaint list to check for complaint status which should be updated by the anti-ragging squad.
- After the enquiry process if any decision is taken accordingly, the student will be able to see what kind of decision/punishment would be taken for the alleged person or persons.

B. **Anti-Ragging Squad Part:**

- The squad member has to log in in accordance to show the complaint list of all complainants.
- After logging in, the squad member can update status for showing the complainants.
- If investigation of anti-ragging squad is found to be sufficient to take action against the alleged accused, necessary actions should be taken accordingly so that the student of that complaint can be able to see that.

2.2 Software Requirement Specification:

2.2.1 Purpose:

The main purpose of this project is to build an anti-ragging complaint system for the students who faces various unwanted incidents at the university premises as well as an interface for the action takers or the decision makers (in this case it is the anti-ragging squad).

2.2.2 Scope:

- Identifying the number of ragging occurrence at a certain period of time in university premise.
- Making a suitable web portal for students to lodge complaint against the alleged person anytime they want.
- Taking necessary actions on the basis of complaints by a group of dedicated members.
- Students can be updated and get information regarding the case by the same portal they are using.
- Higher authority can take steps observing the results/reports on the number of ragging offenses occur or taken care of and pending cases as well.

2.2.3 Hardware Interface:

- Windows
- A browser which supports JSP, HTML, Javascript.

2.2.4 Software Interface:

- **Operating System**: We have chosen windows operating system for its best support and user-friendliness.

- **Development Environment**: We have chosen *NetBeans IDE 8.1*, an open source integrated development environment which supports all kinds of Java applications. It runs on Windows, macOS, Linux and has extensions for languages like JSP, HTML, CSS, Javascript and more.

- **JDK-8:** Java SE Development Kit (JDK) is required to install *NetBeans IDE*. Since we are planning to use Java features in the project, JDK is necessary.

- **Database**: To save all the records of student registration, complaints and actions we have used MySQL database, which is an open source RDBMS.

- **Server**: Developing a java project needs a server. Here, we have chosen GlassFish server which is installed simultaneously with *NetBeans IDE*. It is a lightweight yet powerful application server which can be used as a Web server (Http Server).

2.2.5 Security Requirements:

Security systems need database storage just like many other applications. The student portal should not get all the information gathered and analyzed by the supervising team where as the monitor group should know all the detailed documents regarding the case of each student and that must be taken care of while creating the databases for the application.

CHAPTER – 3

SYSTEM DESIGN

3.1 System Design:

3.1.1 Architecture of Struts MVC Framework:

Struts-based web applications are built to be easily modifiable and maintainable, and the internationalization and flexibility of design is deeply rooted. It relies on standard technologies such as Java beans, Java servlets, Java Server Pages (JSP), and XML. Struts encourages application architectures based on the Model 2 approach, which is a variation of the model-view-controller (MVC) design pattern.

A strict separation is enforced between **processing logic** and **presentation logic** (the **Model** and **View** in Struts parlance), which has a natural consequence of facilitating component reuse.

The centerpiece of Struts is its MVC-style controller, which integrates with other technologies that provide the model and the view. For the model, Struts can interact with standard data access technologies such as JDBC and EJB. For the view, Struts works well with JSP, including the JSP Standard Tag Library (JSTL) and Java Server Faces (JSF), as well as Velocity Templates, XSLT, and other presentation systems.

Table 1 summarizes the three main components of MVC.

	Purpose	**Description**
Model	Maintain data	Business logic plus one or more data sources such as a relational database.
View	Display all or a portion of the data	The user interface that displays information about the model to the user.
Controller	Handle events that affect the model or view	The flow-control mechanism means by which the user interacts with the application.

Table 1: Summary of MVC components

In this three-tier architecture, a JSP page and a Java bean are on an application server, and a data store and the business logic are on a data server: -

1. The browser sends a request to a JSP page.
2. The JSP page communicates with a Java bean.
3. The Java bean is connected to a database.
4. The JSP page responds to the browser.

3.1.2 The Struts Implementation of Model 2:

The Struts implementation of Model 2 uses a specific type of servlet, called an action servlet, and one or more *actions* and action mappings to implement the controller. It also uses a specific type of Java bean, called a *form bean*. As illustrated in Figure 3, the web server at run time contains both the view and controller components of a Model 2 web application, while a third tier (which is usually outside of the web server) contains the model.

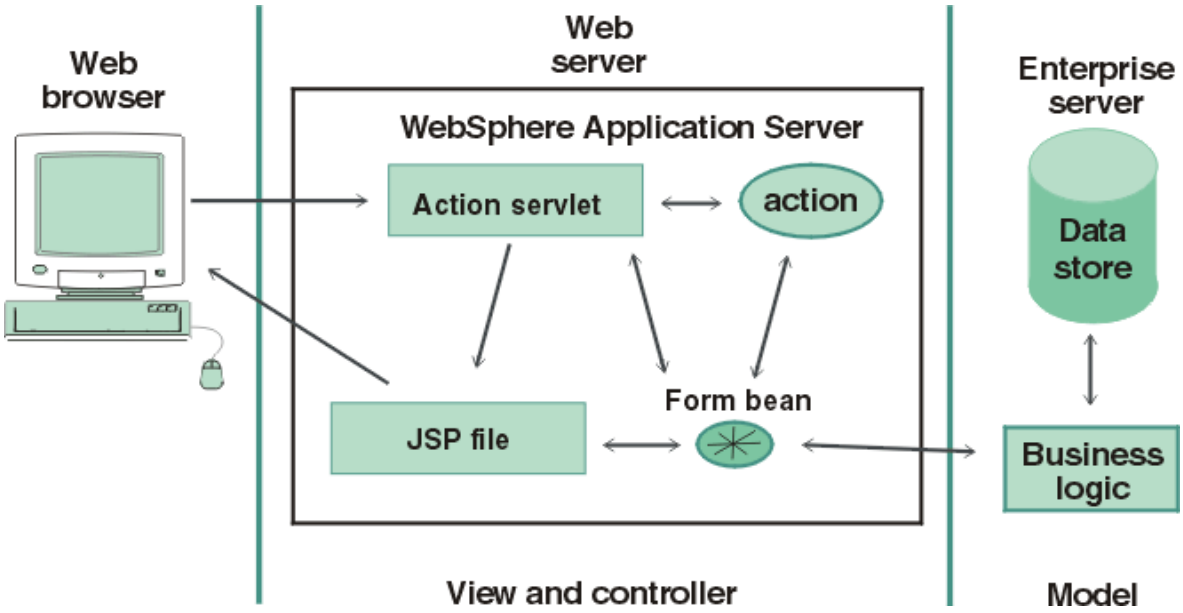


Figure 1 : Structure of Model 2 of Struts

Struts contribution to MVC components are shown in the below Table 2:-

Component	Contribution
Model	None directly. However, the Struts actions and configuration file provide an elegant way to control the circumstances under which the model components are invoked.
View	Java class org.apache.struts.action.ActionForm, which you subclass to create a form bean that is used in two ways at run time: ◦ When a JSP page prepares the related HTML form for display, the JSP page accesses the bean, which holds values to be placed into the form. Those values are provided from business logic or from previous user input. ◦ When user input is returned from a web browser, the bean validates and holds that input either for use by business logic or (if validation failed) for subsequent redisplay. Numerous custom JSP tags that are simple to use but are powerful in the sense that they hide information. Page Designer does not need to know much about form beans, for example, beyond the bean names and the names of each field in a given bean.
Controller	The Struts action servlet handles runtime events in accordance with a set of rules that are provided at deployment time. Those rules are contained in a Struts configuration file and specify how the servlet responds to every outcome received from the business logic. Changes to the flow of control require changes only to the configuration file. Struts also provides the Java class org.apache.struts.action.Action, which a Java developer subclasses to create an "action class". At

Component	Contribution
	run time, the action servlet is said to "execute actions," which means that the servlet invokes the execute method of each of the instantiated action classes. The object returned from the execute method directs the action servlet as to what action or JSP file to access next. To facilitate reuse, invoke business logic from the action class rather than including business logic in that class.

Table 2. Struts contributions to model, view, and controller

3.1.3 Work Flow of Struts 1.x Framework:

The following events happen when the Client browser issues an HTTP request.

- The ActionServlet receives the request.
- The struts-config.xml file contains the details regarding the Actions, ActionForms, ActionMappings and ActionForwards.
- During the startup the ActionServelet reads the struts-config.xml file and creates a database of configuration objects. Later while processing the request the ActionServlet makes decision by refering to this object.

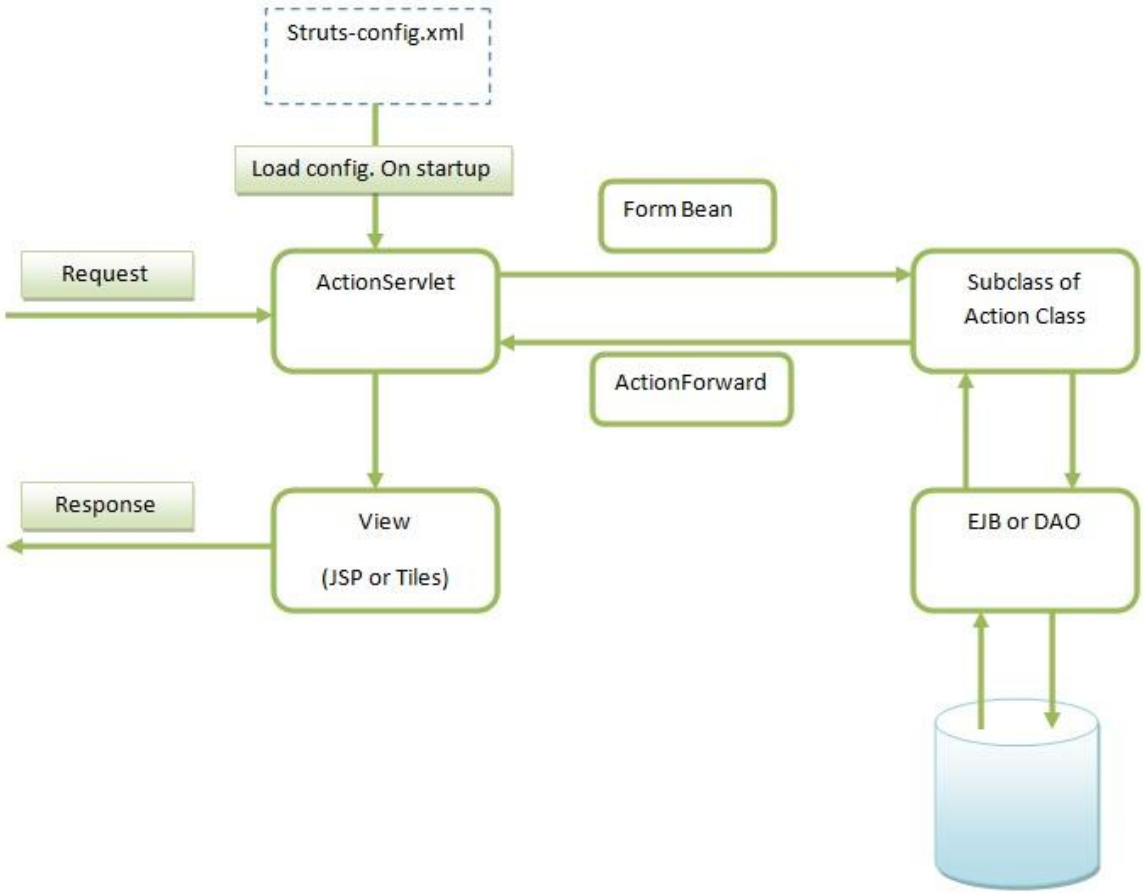


Figure 2: Architecture of Struts 1.x

When the ActionServlet receives the request, it does the following tasks.

- Bundles all the request values into a JavaBean class which extends Struts ActionForm class.
- Decides which action class to invoke to process the request.
- Validate the data entered by the user.
- The action class process the request with the help of the model component. The model interacts with the database and process the request.
- After completing the request processing the Action class returns an ActionForward to the controller.

- Based on the ActionForward the controller will invoke the appropriate view.
- The HTTP response is rendered back to the user by the view component.

CHAPTER – 4

IMPLEMENTATION

4.1 Overview of the Application:

When we use Struts, the framework provides us with a controller servlet, ActionServlet, which is defined in the Struts libraries that are included in the IDE, and which is automatically registered in the web.xml deployment descriptor.

The controller servlet uses a struts-config.xml file to map incoming requests to Struts Action objects and instantiate any ActionForm objects associated with the action to temporarily store form data. The Action object processes requests using its execute method, while making use of any data stored in the form bean. Once the Action object processes a request, it stores any new data (i.e., in the form bean, or in a separate result bean), and forwards the results to the appropriate view.

Developing a Struts application is similar to developing any other kind of web application in NetBeans IDE. However, we complement your web development toolkit by taking advantage of the Struts support provided by the IDE. For example, we use templates in the IDE to create Struts Action objects and ActionForm beans. Upon creation, the IDE automatically registers these classes in the struts-config.xml file and lets us extend this file very easily using menu items in the Source Editor's right-click menu. Because many web applications use JSP pages for the view, Struts also provides custom tag libraries which facilitate interaction with HTML forms. Within the IDE's Source Editor, we can invoke code completion and Javadoc support that helps you to work efficiently with these libraries.

4.2 Setting Up the Application:

In the IDE, we create a Struts application in the following way using the New Web Application wizard:

1. Choose File > New Project from the main menu. Select Java Web in the list of Categories and then select Web Application in the list of Projects. Click Next.
2. In the Name and Location panel, enter Anti-Ragging Portal for Project Name and click Next.
3. In the Server and Settings panel, select the server to which we want to deploy your application. In this case, it is GlassFish Server 4.1.1. The Context Path to your deployed application becomes /Anti-Ragging Portal. Click Next.
4. The wizard displays the following configuration options.
    - **Action Servlet Name**: The name of the Struts action servlet used in the application. The web.xml deployment descriptor contains an entry for the action servlet and specifies the appropriate parameters.
    - **Action URL Pattern**: Specifies the patterns of incoming requests mapped to the Struts action controller. This generates a mapping entry in the deployment descriptor. By default, only the *.do pattern is mapped.
    - **Application Resource**: Lets us specify the resource bundle which will be used in the struts-config.xml file for localizing messages.

.

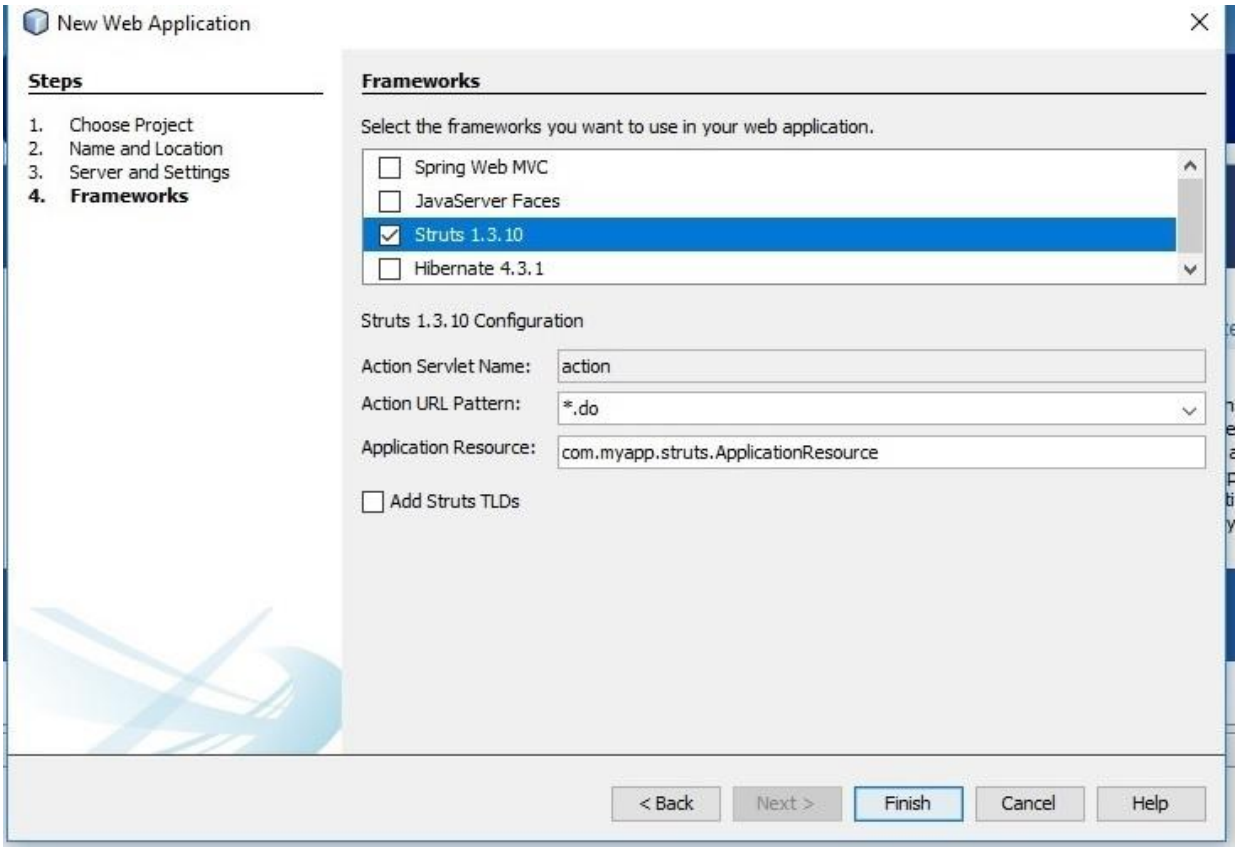


Figure 3: Wizard for creating new project

By default, this is com.myapp.struts.ApplicationResource.

- **Add Struts TLDs**: Lets us generate tag library descriptors for the Struts tag libraries. A tag library descriptor is an XML document which contains additional information about the entire tag library.

5. Click Finish. The IDE creates the project folder in the file system. As with any web application in the IDE, the project folder contains all of your sources and the IDE's project metadata. Application has all of the Struts

libraries on its classpath. They are included in the project and will be packaged with it later when you build the project.

4.3 Creating JSP Pages:

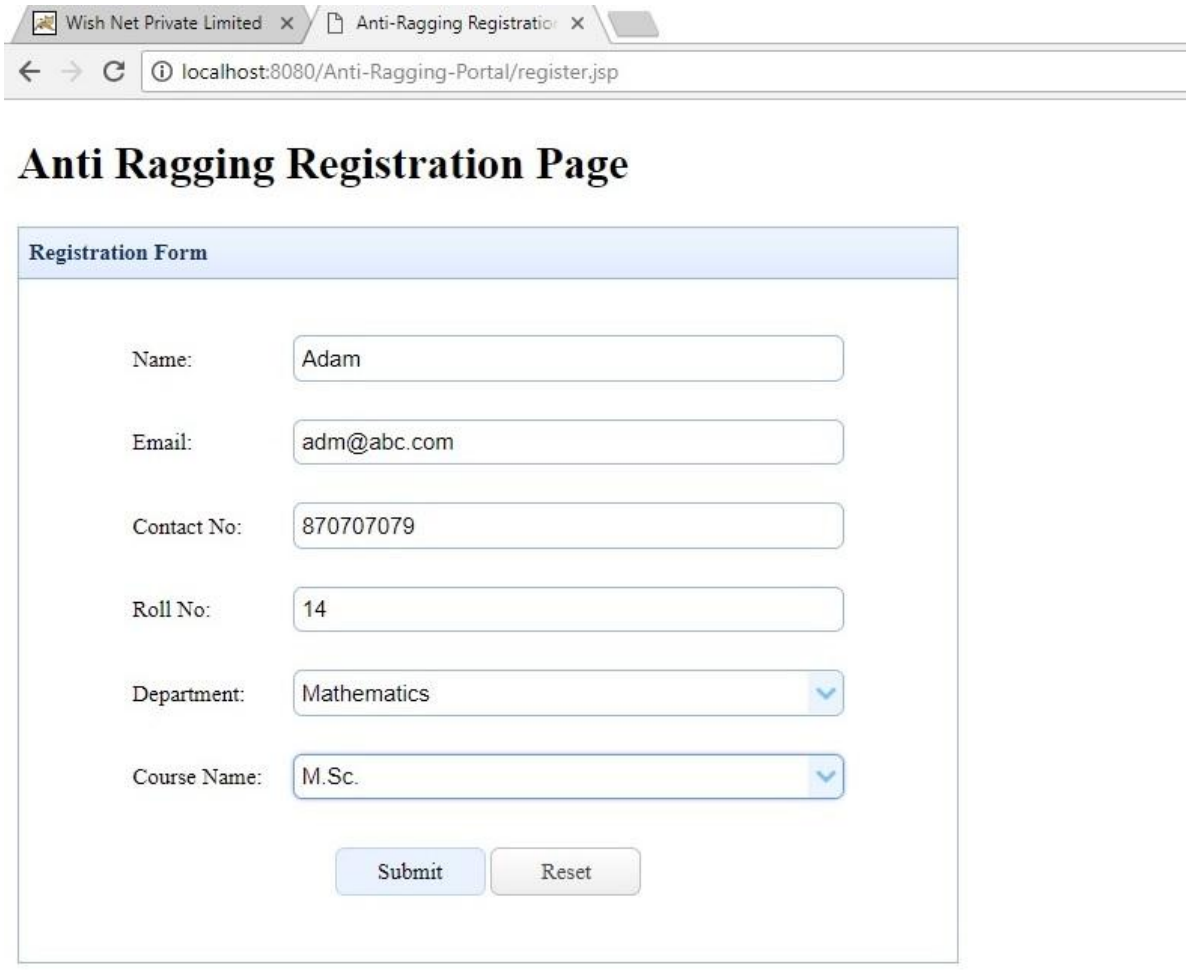
The way to create a jsp page inside the Anti-Ragging Portal project is as follows:

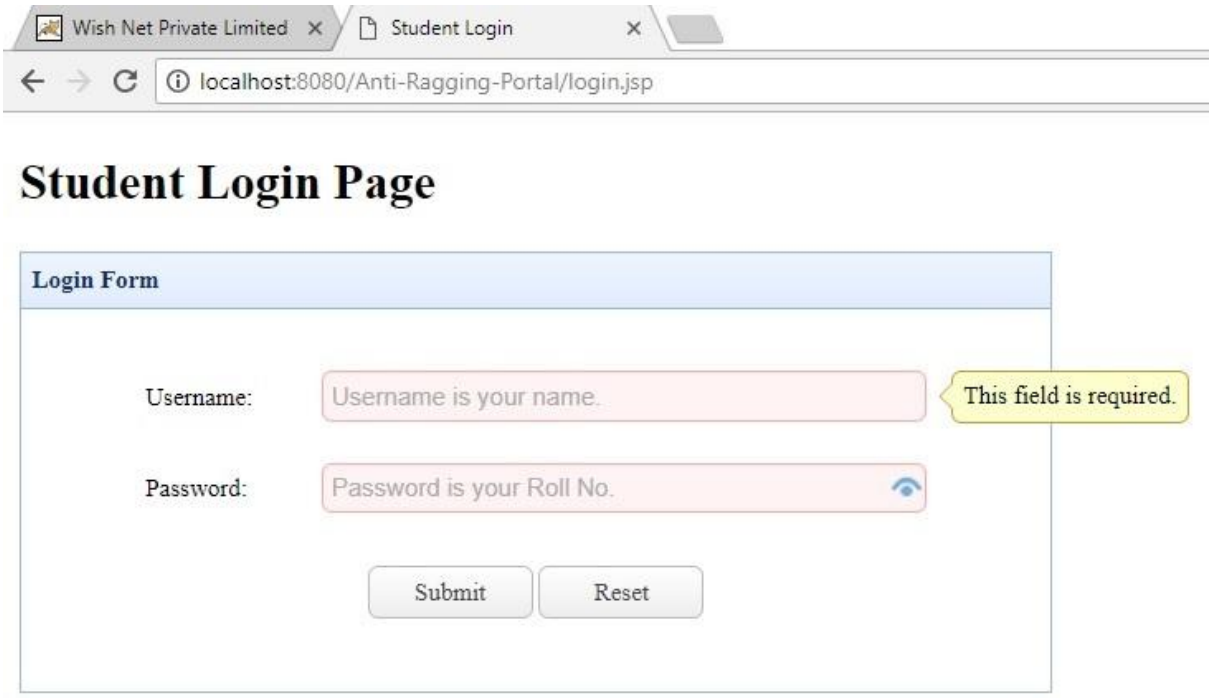
- Right-click the Anti-Ragging Portal project node, choose New > JSP
- name the new file as you want to name it
- Click Finish
- The jsp file opens in the Source Editor.

In this application, there are many jsp pages, which includes login pages, form pages, success and failure pages. Some of those pages are shown sequentially to understand the project.

- The Registration page for students to register before lodging any complaint is the first job for them.
- And they have to login each time they want to complaint or check for any status on the basis of the complaint/complaints they already lodged.

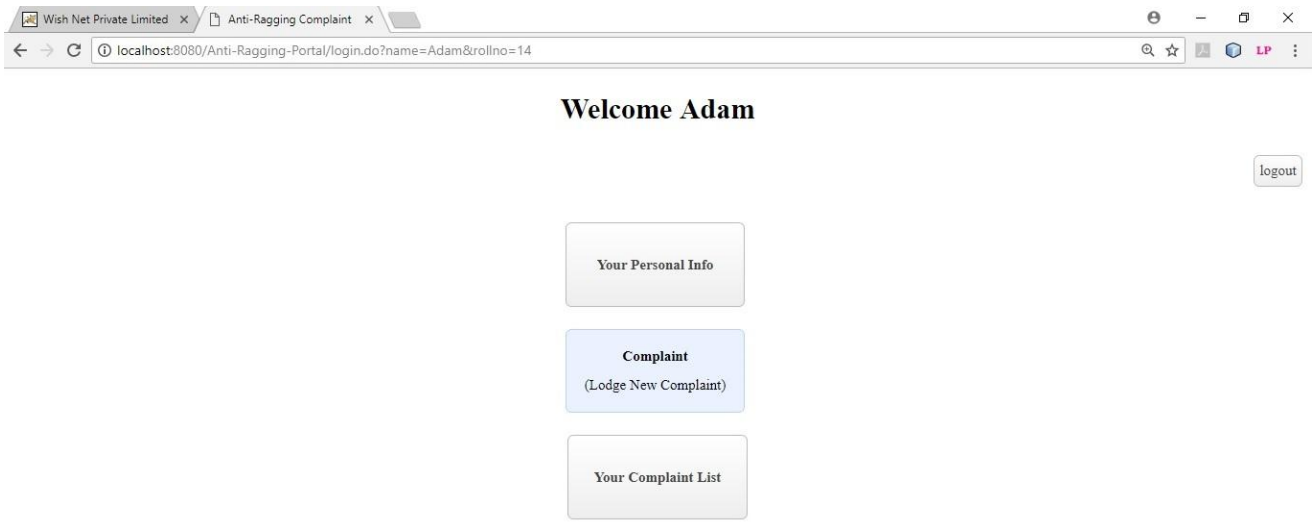
The view of the registration and login pages are shown on the next page:



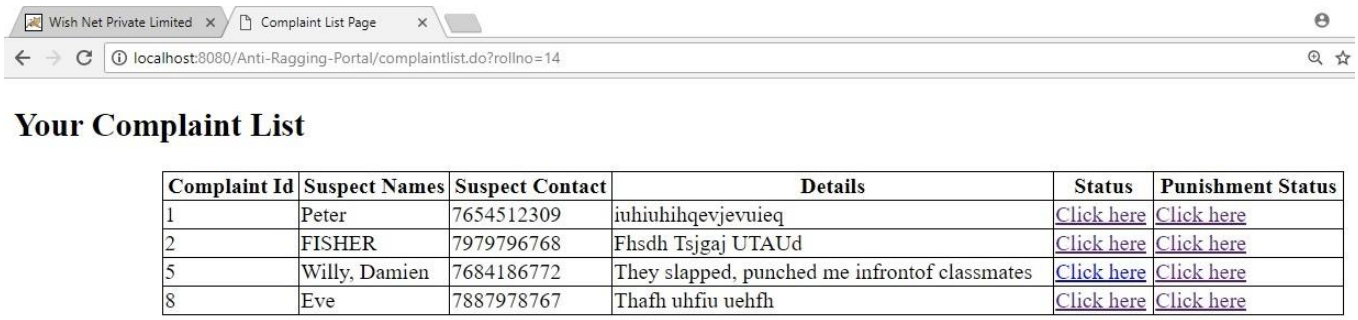


- Now, once a student logged in to his/her profile, there are link buttons for them to see personal info, lodge complaint and show complaint list.
- If they click the “lodge Complaint” button, they will see the complaint form which they need to fill up to complaint successfully.
- And they will see their complaint details and status which needs to be updated by the anti-ragging squad.

Student Profile view is shown here:

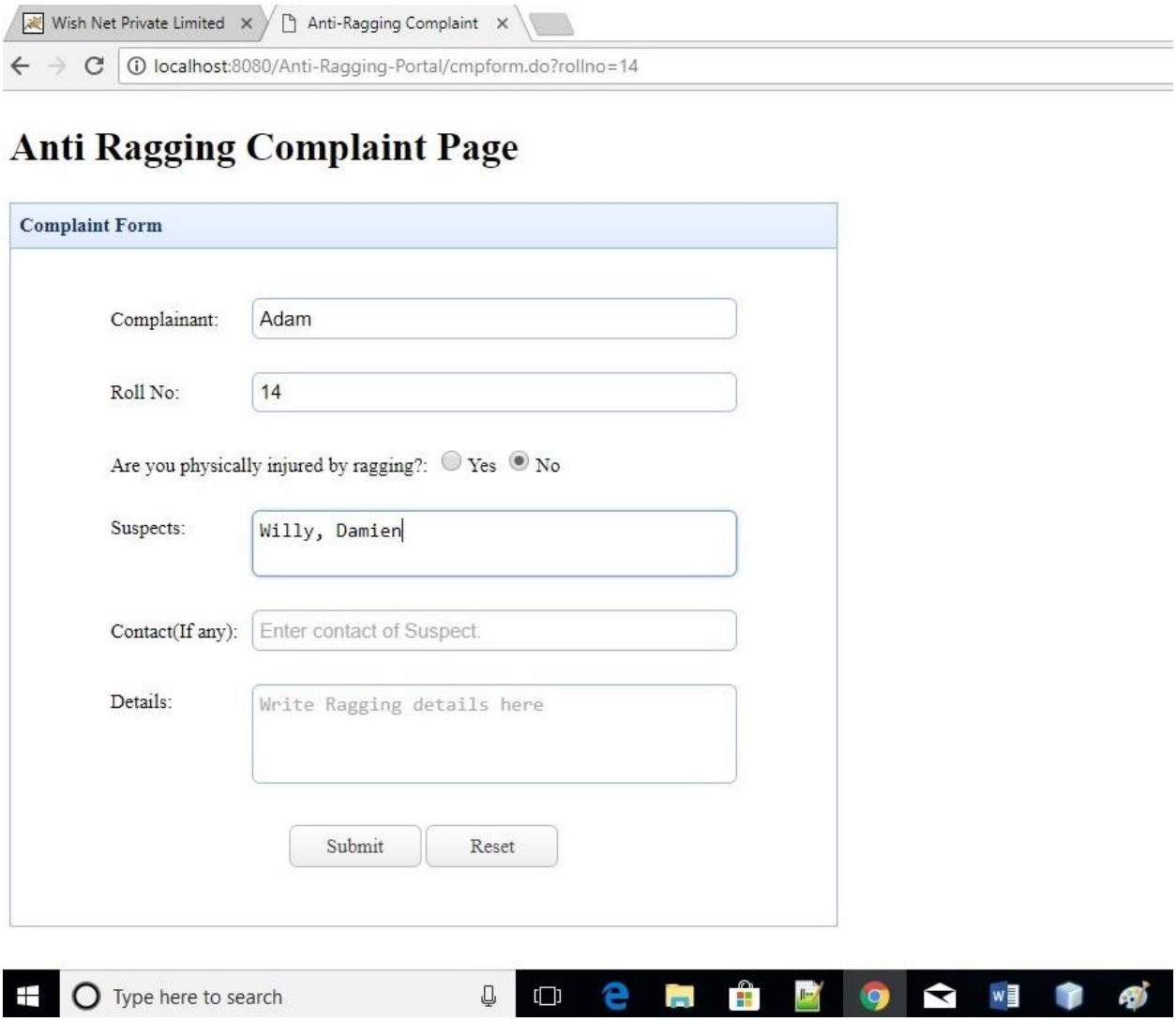


Below are the complaint list of Adam (Student): -



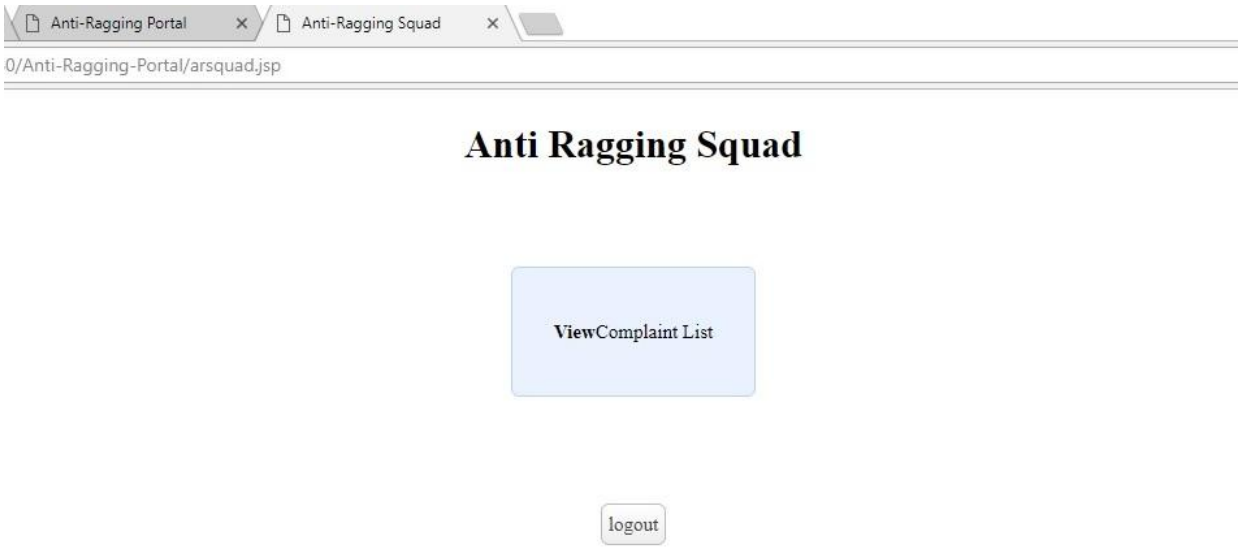
Complaint Id	Suspect Names	Suspect Contact	Details	Status	Punishment Status
1	Peter	7654512309	iuhiuhihqevjevuieq	Click here	Click here
2	FISHER	7979796768	Fhsdh Tsjgaj UTAUd	Click here	Click here
5	Willy, Damien	7684186772	They slapped, punched me infrontof classmates	Click here	Click here
8	Eve	7887978767	Thafh uhfiu uehfh	Click here	Click here

The complaint form view is created as: -

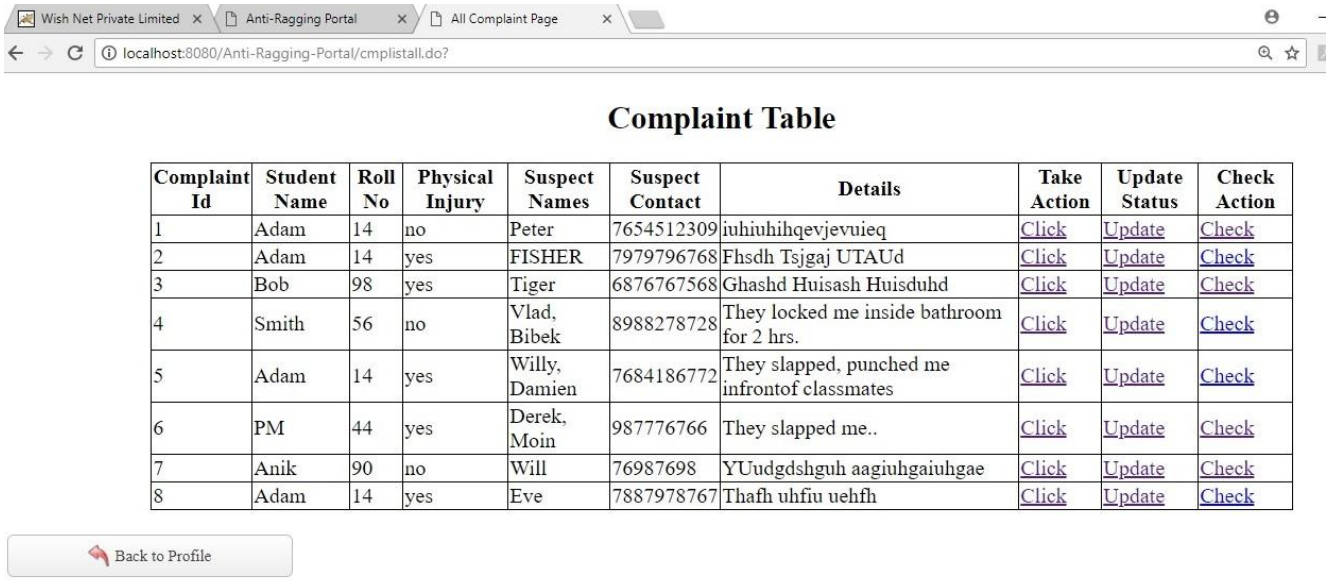


Anti-ragging complaint form

- Now, the anti-ragging squad portal starts with a login page for them which has a unique username and password for them to login.
- After logging in to the portal, they can view the complaints of all students containing unique complaint id for each case.



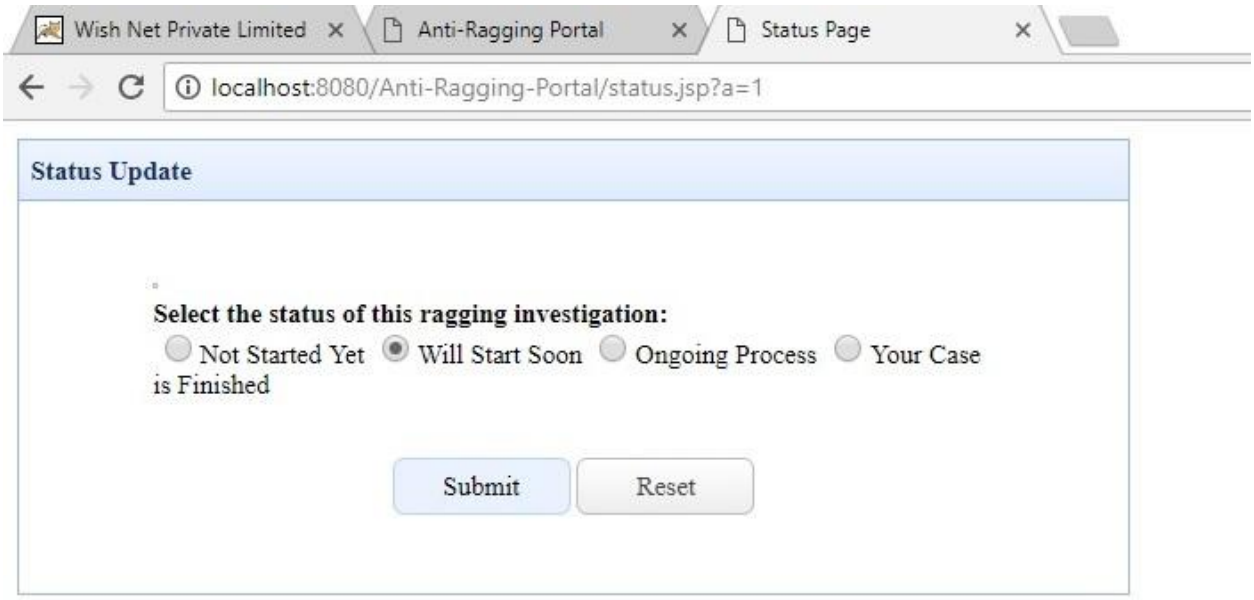
The complaint list for anti-ragging squad is as follows:

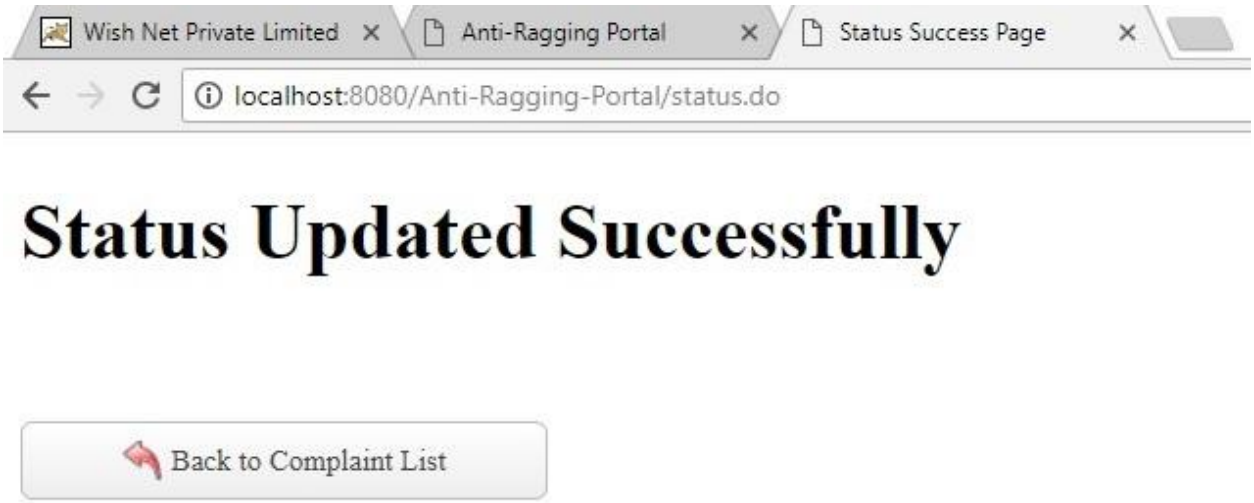


Complaint Id	Student Name	Roll No	Physical Injury	Suspect Names	Suspect Contact	Details	Take Action	Update Status	Check Action
1	Adam	14	no	Peter	7654512309	iuhiuhihqevjevuieq	Click	Update	Check
2	Adam	14	yes	FISHER	7979796768	Fhsdh Tsjgaj UTAUd	Click	Update	Check
3	Bob	98	yes	Tiger	6876767568	Ghashd Huisash Huisduhd	Click	Update	Check
4	Smith	56	no	Vlad, Bibek	8988278728	They locked me inside bathroom for 2 hrs.	Click	Update	Check
5	Adam	14	yes	Willy, Damien	7684186772	They slapped, punched me infrontof classmates	Click	Update	Check
6	PM	44	yes	Derek, Moin	987776766	They slapped me..	Click	Update	Check
7	Anik	90	no	Will	76987698	YUudgdshguh aagiuhgaiuhgae	Click	Update	Check
8	Adam	14	yes	Eve	7887978767	Thafh uhfiu uehfh	Click	Update	Check

- While analyzing the cases, the squad would have different status for different cases and they should let the student know the status of their complaints. This is sorted by the column “Update Status”.

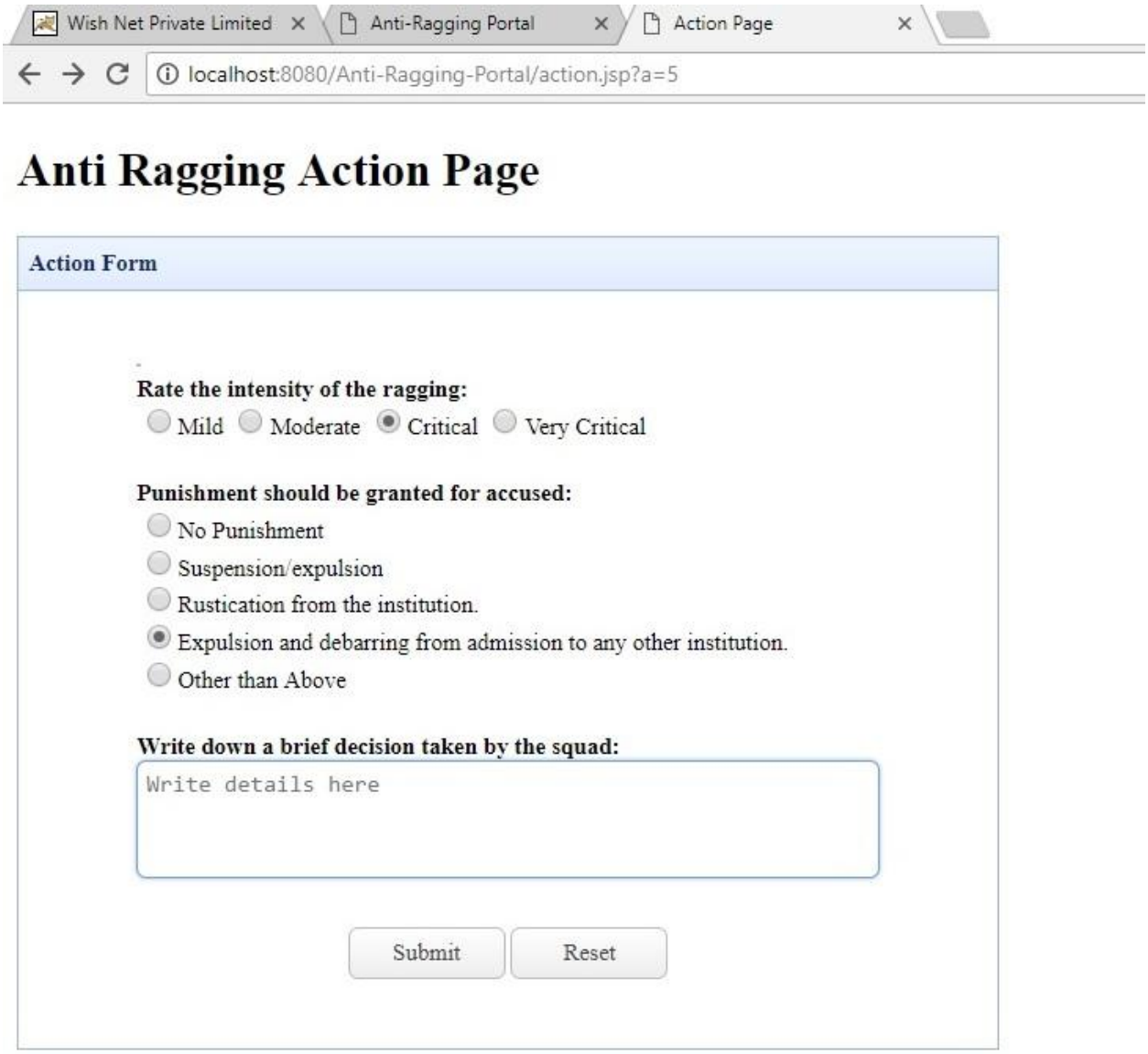
Let’s see what happens by clicking the “Update Status” column:

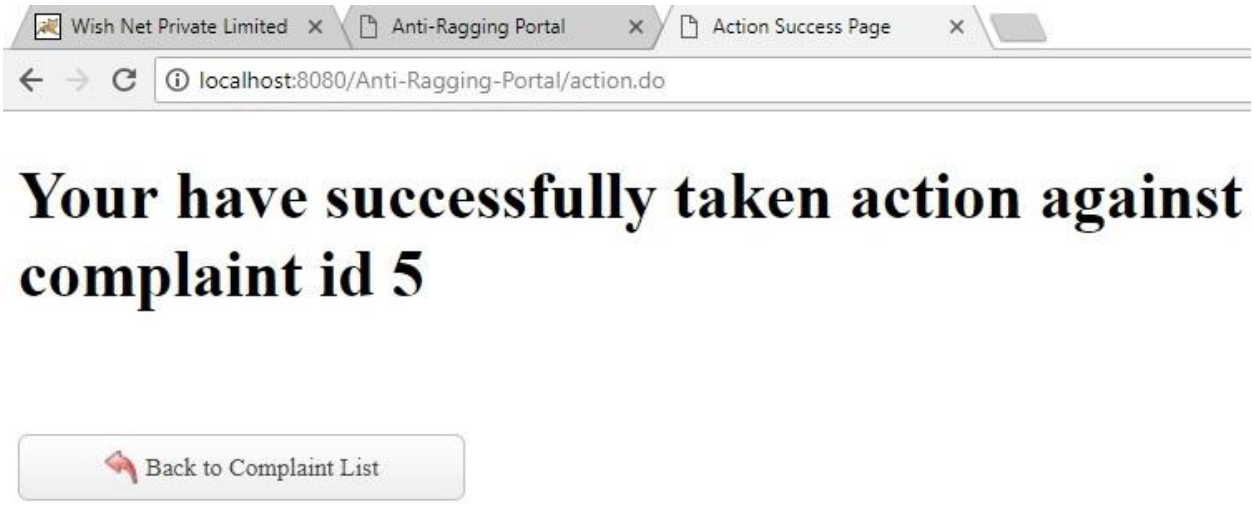




- As shown in the Complaint List Table, they can see all the complaints and analyzing the cases they take their take decision accordingly by clicking on “Take Action” column.

Let’s see what happens by clicking “Take Action” column:

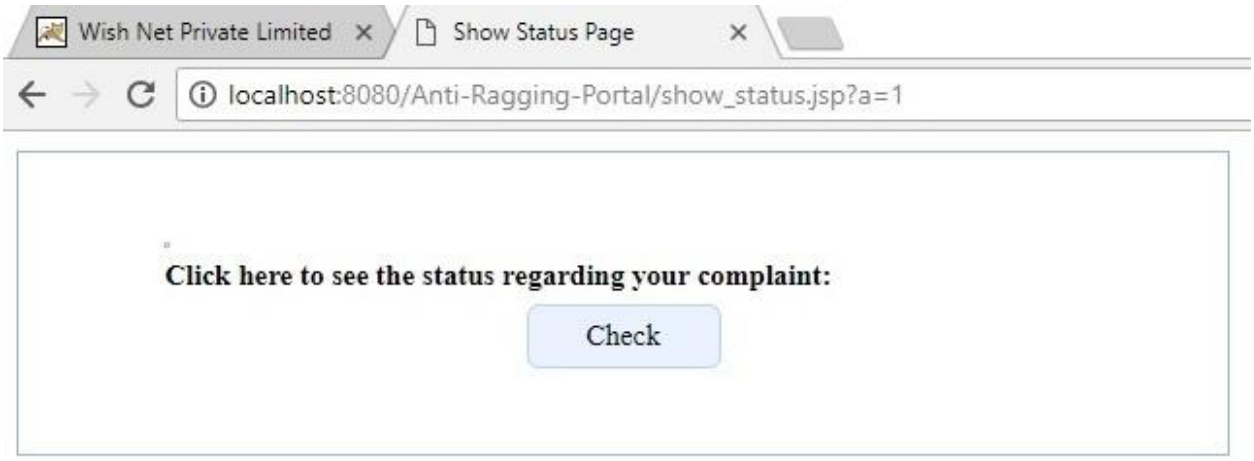

Wish Net Private Limited
Anti-Ragging Portal
Action Page
localhost:8080/Anti-Ragging-Portal/action.jsp?a=5
Anti Ragging Action Page
Action Form
Rate the intensity of the ragging:
Mild
Moderate
Critical
Very Critical
Punishment should be granted for accused:
No Punishment
Suspension/expulsion
Rustication from the institution.
Expulsion and debarring from admission to any other institution.
Other than Above
Write down a brief decision taken by the squad:
Write details here
Submit
Reset



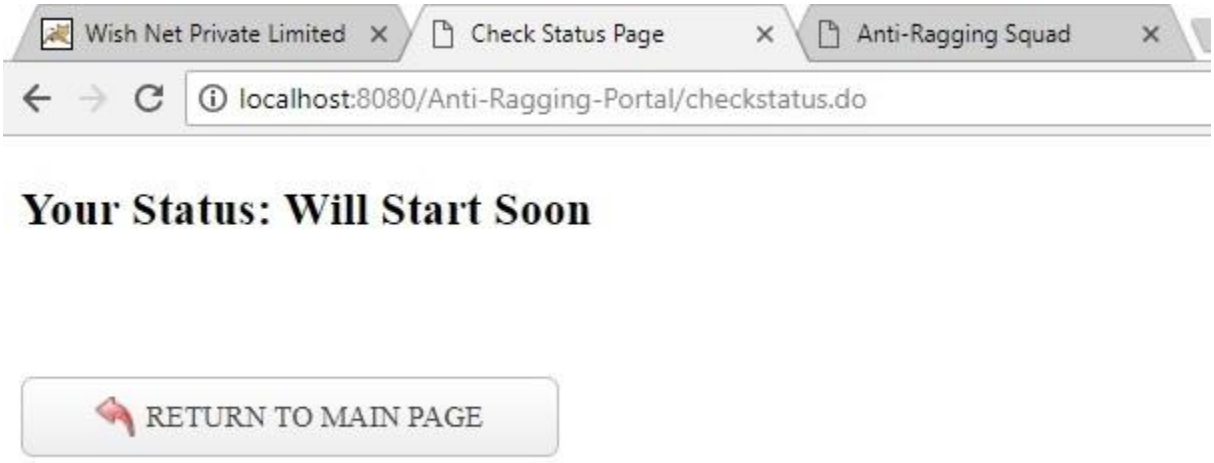
- If a squad member clicks on the "Check Action" column or mistakenly try to take action for a complaint id which is already taken, then he/she will face this:

- Now, back to student profile and we see how a student gets to know his status:

This page is seen by clicking "Status" by the student:



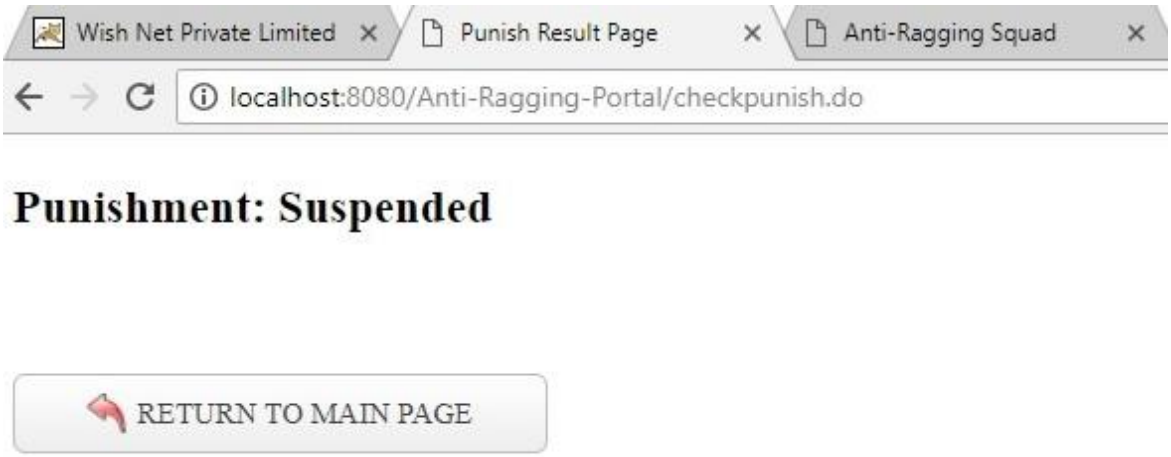
Once he presses the check button, he will see his status updated by anti-ragging squad member:



- When the student sees in the status that "The case is finished", he will get to check the punishment granted by the authority and only that. Other detailed information will be hidden from the student.

Let's see how it works:

Pressing the check button goes to the page shown on the next page:



4.4 Creating ActionForm Bean:

A Struts ActionForm bean is used to persist data between requests. For example, if a user submits a form, the data is temporarily stored in the form bean so that it can either be redisplayed in the form page (if the data is in an invalid format or if login fails) or displayed in a login success page (if data passes validation).

In this project, we have used two ActionForm Bean classes mainly to declare all the variables used in the application and define all the setter, getter methods needed for each variable.

- Right-click the project node and choose New > Other. Under Categories choose Struts, then under File Types choose Struts ActionForm Bean. Click Next.
- Type in ComplaintForm for the Class Name. Then select com.myapp.struts in the Package drop-down list and click Finish.
- The IDE creates the ComplaintForm bean and opens it in the Source Editor. By default, the IDE provides it with a String called name and an int called number. Both fields have accessor methods defined for them. Also, the IDE adds a bean declaration to the struts-config.xml file. If we open the struts-config.xml file in the Source Editor.
- After declaring variables, Right Click on one of the variable name and select Refactor > Encapsulate Fields:

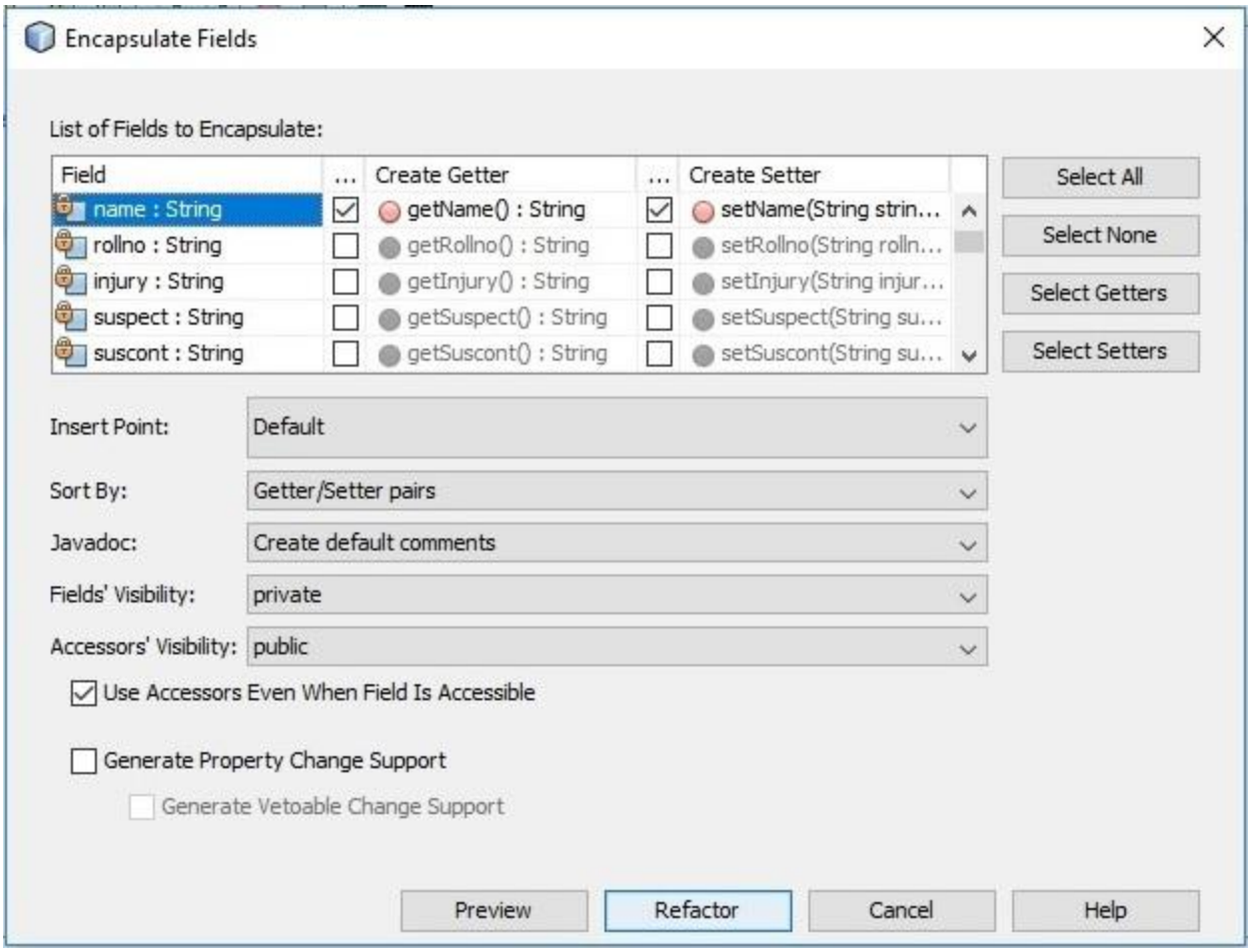


Figure 4: Encapsulate Fields and Refactor

4.5 Creating Action & DAO Classes:

Action classes are defined to handle requests. Actions exists between the Model and View of an application. The struts-config.xml file designates

the Action classes that handle requests for various URLs. The Action objects do:

- Invoke the appropriate business
- Data-access logic
- Store the results in FormBean
- Designate the type of situation (missing data, database error etc.)

For this project, multiple action and Dao classes are created within Action and DAO packages. The process to create action classes is shown below:

- In the Projects window, right-click the project node and choose New > Other. From the Struts category choose Struts Action and click Next.
- In the Name and Location panel, change the name to ComplaintAction.
- Select com.myapp.struts in the Package drop-down list.
- Type /complaint in Action Path. Make sure settings appear as in the screenshot below, then click Next.

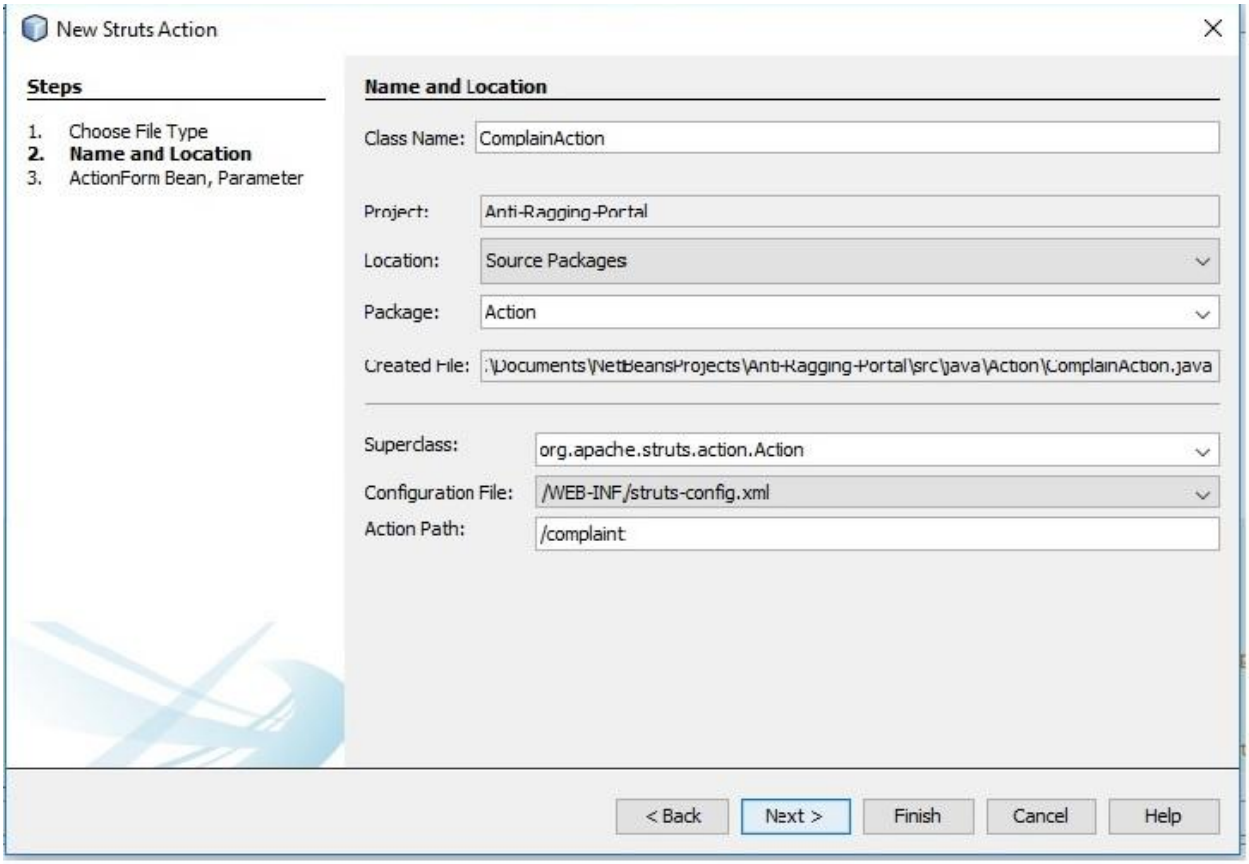


Figure 5: Wizard for creating Action class

- In the next step of the wizard, we are given the opportunity to associate the Action class with a form bean. The ComplaintForm bean we previously created is listed as an option for ActionForm Bean Name. Then we need to make the following adjustments to the panel:
  - Delete the forward slash for the Input Resource field
  - Set Scope to Request (Session is the default scope setting in Struts.)
  - Deselect the Validate ActionForm Bean option

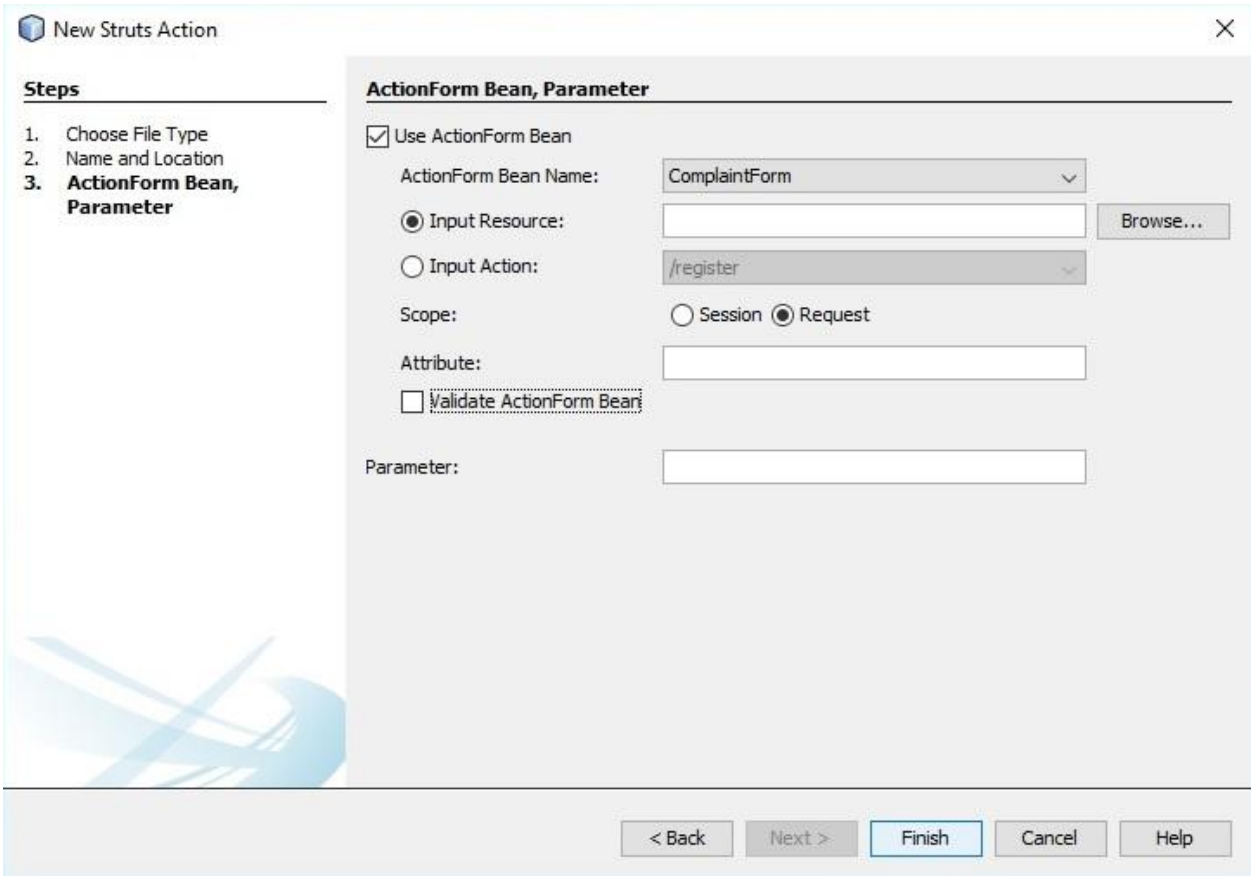


Figure 6: End step to create Action class

After clicking Finish, ComplaintAction class is generated, and the file opens in the Source Editor. Also, the following action entry is added to the struts-config.xml file.

- DAO classes are used in Struts to access database and extract data from there separately from the Action classes. The Struts Action class provides us with the reference to the form.
- In this project, we want to use it in a DAO and pass it to the DAO as parameter. We created a package called DAO and within it we have put all the dao classes. In these classes, we have defined the business methods required connecting the databases.

- All the insertion, deletion, updatation, manipulation of database data are implemented within the business methods of DAO classes and they are called from different Action classes.

4.6 Implementing Validation:

There are two types of validation in Struts that can be implemented, those are:

- Pre-Validation
- Post-Validation

- Pre-Validation: We have used EasyUI forms and javascript for the pre-validation is provided by EasyUI. It is complete framework to build modern, interactive, javascript application.

- Post-Validation: We can call it the Struts validation which is implemented as follows:

  - Struts validation is implemented within the execute method of Action classes.
  - In order to use incoming form data, we need to take execute's ActionForm argument and cast it as the name of the FormBean class.
  - Then, we need to apply the getter methods we created earlier in the FormBean.
  - Now, we need to type condition clauses to perform validation on the incoming data.

4.7 Configuring & Running the Application:

- The IDE uses an Ant build script to build and run the web application. The IDE generated the build script when we created the project, basing it on the options entered in the New Project wizard. Before we build and run the application, you need to set the application's default entry point to the jsp page we wish to see first.

- Whatever the jsp file name we use for this, it must be listed within the welcome tag in web.xml file like:

```
<welcome-file>index.jsp</welcome-file>
```

- Now we need to attach stylesheet provided by EasyUI within the project. We can simply copy and paste all the css files inside the web folder of the project and link those files to the jsp pages.

- In the Projects window, right-click the project node and choose Run. The IDE builds the web application and deploys it, using the server we specified when creating the project. The browser opens and displays the index.jsp page. Now, we can perform operations as already discussed in the section 4.3.

CHAPTER – 5

CONCLUSION & FURTHER WORK

5.1 Conclusion:

This concludes the report on the Development of Anti-Ragging Portal application. This document demonstrates the requirements to develop this web application as well as the procedure to implement the project successfully. Ragging is undoubtedly a big and growing issue specially in universities and colleges. To control these type of unwanted phenomena, a university or a college should have a properly runnable and well-structured portal handled by efficient persons. So that, whenever a student faces these incidents they can report as fast as possible to the authorized body without any mail or visit which is time consuming. And this application in a way shows how we can fight ragging by providing a good web application to the students. Not only that, the student gets the update of his complaints status without any phone call or queries. Finally, after a proper investigation, the student will be able to see the information that he should know regarding the decision taken by the governing body, which is satisfying to the complainant as well as to the squad body.

5.2 Further Improvements:

Further improvements can be done to the application discussed in this report which are listed below:

- Apart from always logging in and check for status, the student can get a confirmation mail after registration or lodging any complaint.

- The mail should go to the anti-ragging squad after any student lodge a complaint so that, they can take action more quickly.

- The suspect/ suspects should get a notice on the complaint lodged against him/ her from the authority by phone call or mail.
- The design of the front end may be improved by using more user-friendly interface.

- In the back-end we have used different Action classes, but it can be more efficient if we punch all the Action classes using Dynamic Dispatch which have only one Action class to perform all the actions.

REFERENCES

1. https://netbeans.org/kb/docs/web/

2. http://www.javawebtutor.com/articles/struts/

3. https://coderanch.com

4. https://www.roseindia.net/tutorialsearch/

5. http://www.mkyong.com/tutorials/struts-tutorials/

6. http://www.raistudies.com/struts-1/

7. https://dzone.com/tutorials/java/struts/

8. https://www.ibm.com/support/knowledgecenter/

9. www.wideskills.com/struts/