**LAB MANUAL**

**FOR**

**ENTERPRISE COMPUTING IN JAVA**

**Submitted By: Aparna Ramamurthy**

**SOIT**

**Practical #1**

Objective: Getting started with Servlets

**Assignment No.1**

1. Install JDK
2. Install Eclipse
3. Install Tomcat
4. Set the necessary environment variables to run Eclipse and Tomcat for creating J2EE applications.
5. Add Tomcat to Eclipse.
6. Create a new Dynamic Web project.

Create the following Servlet Applications:

1. Create a Servlet to print the location of vowels in a String.
2. Write a method which calculates the number of definite holidays (Saturdays and Sundays) in a month, given the number of days in the month and the start day of the month as input.  
   Example :

31 Wednesday

Number of holidays is 8.

1. Print a number grid: Input a grid size (number rows and columns) from user. Based on the grid size create a grid of numbers (1-9). The numbers in the cells will come randomly.
2. Either 1 of Snake and Ladder and Pattern Matching problems provided to you.

**Theory and Concept**

**Practical #2 Request Handling**

**Objective:- To understand how various form requests are handled.**

1. Create a Login form which has a username and password. On Login the user should be authenticated and the system should print the user name and password and print if the user is a student or a faculty.  
     
   Pre-requisites: Create a file data which will have a list of 8 users – 5 students and 3 faculties. The authentication should be done against this file data. Call this user file.  
     
   Tips: Ensure that the file data is loaded only once.
2. Create a page so that a student can select 3 or more subjects that he wants to do in a particular course. Assume there are 2 courses MBA and MCA and student will belong to only one course. Each course will have 5 subjects of which students can choose two or more.

The student will enter his name, email address, enrollment number, course he belongs to and the subjects he wants to choose. Add the entered data to a file by name course-file and also display the results on the browser.

1. Create a page in which a teacher can enter the marks of students in every subject of a particular course. There will be 2 types of courses MCA and M.tech and 2 subjects in each. For simplicity, you may assume that no subject will be in more than 1 course. Each course will have 5 students.  
     
   Pre-requisites : Create a file data for the above problem. Call this course file. The marks should be saved to the course file. You could re-use Q1 and Q2 data.

Output: The marks should be saved to the course file and also be displayed on the screen.

1. Link pages 1 and 2; and 1 and 3 that is, after authentication a teacher/student can enter his details.
2. Create a registration form for students and faculties. The form should have the following fields:
   1. Name
   2. User name (should be unique)
   3. Password
   4. Email
   5. Sex – male/female
   6. Address
   7. Phone
   8. Alternate contact number
   9. Degree details – 10, 12, bachelors, masters (if there)
   10. Type – faculty or student

All the entered data should be accessed in one go and saved to the file user data.

**Theory and Concept**

**Practical #3**

Objective:- To U**nderstand ServletContext and Servlet Config and redirection modes.**

1. Write a program to print the number of hit counts on your application.
2. Write a program to see the number of a times the sports page of times of India is visited.
3. Modify programs 2, 3 and 5 to call another page that will display the output in HTML. What method will you use for this and why?
4. Link 1 and 5 of Assignment 2 – that is if user authentication failed send to registration page. Which redirection mechanism will you use and why?
5. Modify programs 1, 2, 3 and 5 to ensure that the location of the data files are configurable and can be changed easily at any time.

**Theory and Concept**

**Practical .4**

**Objective:-** Understand JDBC and its use in Servlets.

1. Add the database drivers to Eclipse. Describe the steps to do this.
2. Link the programs 1, 2, 3, 4 and 5 (if not already done so) so that they now look like a consolidated application.
3. Modify all the questions of assignments 2 and 3 so that the data is from the database and not flat file.

Here you will first need to create a proper database structure with ER, followed by tables. Then create data into the tables. Describe the steps for each of these. Then modify your servlet codes.

1. Now you need to create a mailing interface so that the faculty and students can send emails to each other. Your mail interface will contain To:, From, Subject and Message. The To for students will be a selection from faculty names. The To for Faculty should be a selection from student names, specific for a course. Note that now these data will come from database.

The mail data will be stored in database. At any given point of time the faculty and students will get to see their Inbox. On refreshing, the latest mails should come from the database. The view rendering for Inbox will be done by a separate view page which will be called once the servlet retrieves mail data for specific users. Explain the steps for these – which method you use here and why.

**Theory & Concept**

**Practical #5 and 6**

**Objective:- To understand Session handling and Exception handling.**

By the time you have reached here you have a single application in place which has two types of users: Students and Faculty. Each can register onto the system and then Login and perform the following functionalities:

Faculties

1. Enter marks of students
2. View marks of students
3. Send mail to students.

Students

1. Choose subjects of a course
2. View their marks
3. Send mail to faculties

If you do not have these in place, ensure this is done.

1. Now, try going to the view marks page without logging in. What happens. Explain.
2. Try adding marks for a student without logging in. Explain what happens.
3. Use Cookies to handle session between Login and add marks page.
4. Now Disable cookies on browser and explain what happens to the session handling you just did.
5. Use HttpSession handling between Login and View.
6. Now Disable cookies on browser and explain what happens to the session handling you just did.
7. Use HttpSession and URL rewriting for pages between Login and Mail.
8. Now Disable cookies on browser and explain what happens to the session handling you just did.
9. Explain what kind of exceptions can occur in the pages you have created.
10. Create a single error handling page to habdle all exceptions and write error handling code.