# LEAVE AND ON DUTY FORM MANAGEMENT SYSTEM

SUPERVISED BY

Mr.K.DINESH KUMAR M.E.

COMPLETED BY

- Nandhagopal R
- Naveen R
- Praveenraj M S
- Madhuvanthi M

# ABSTRACT

The main objective of the proposed system is to automate the leave and on duty form and to decrease the paper work and easier record maintenance by having a particular website for leave and on duty maintenance. This approach basically deals with the record of leave and on duties taken by students in the organization. This system also approach’s to reduce the formalities and time delay faced by the students. The students apply for the leave, faculty checks whether the reason for the leave or on duty is valid or not and if so the reason is valid and the leave is accepted by the faculty otherwise it will be rejected. If the leave is rejected by the faculty, the student has to continue the process from the beginning to apply the leave. In this paper we review the various computerized system which is being developed by using different techniques. Based on this review the **Leave and On Duty Form Management System** simplifies the leave and on duty process, makes it more maintainable, gives better, clearer and more frequent information to end users, standardizes the processing of different types of leave and lowers the amount of data entry and verification activities.

# Existing System

- In most of the organizations, each student has been provided with leave or on duty form at the time of appointment.
- In the existing system, the manual process, receiving data from students and getting approval are done through manual process.
- Students have to suffer lots of problems and formalities for the approval of leave and on duty.
- These records are entered in manual process will take long time, separate workers need to maintaining the databases. All the details are stored via separate databases.

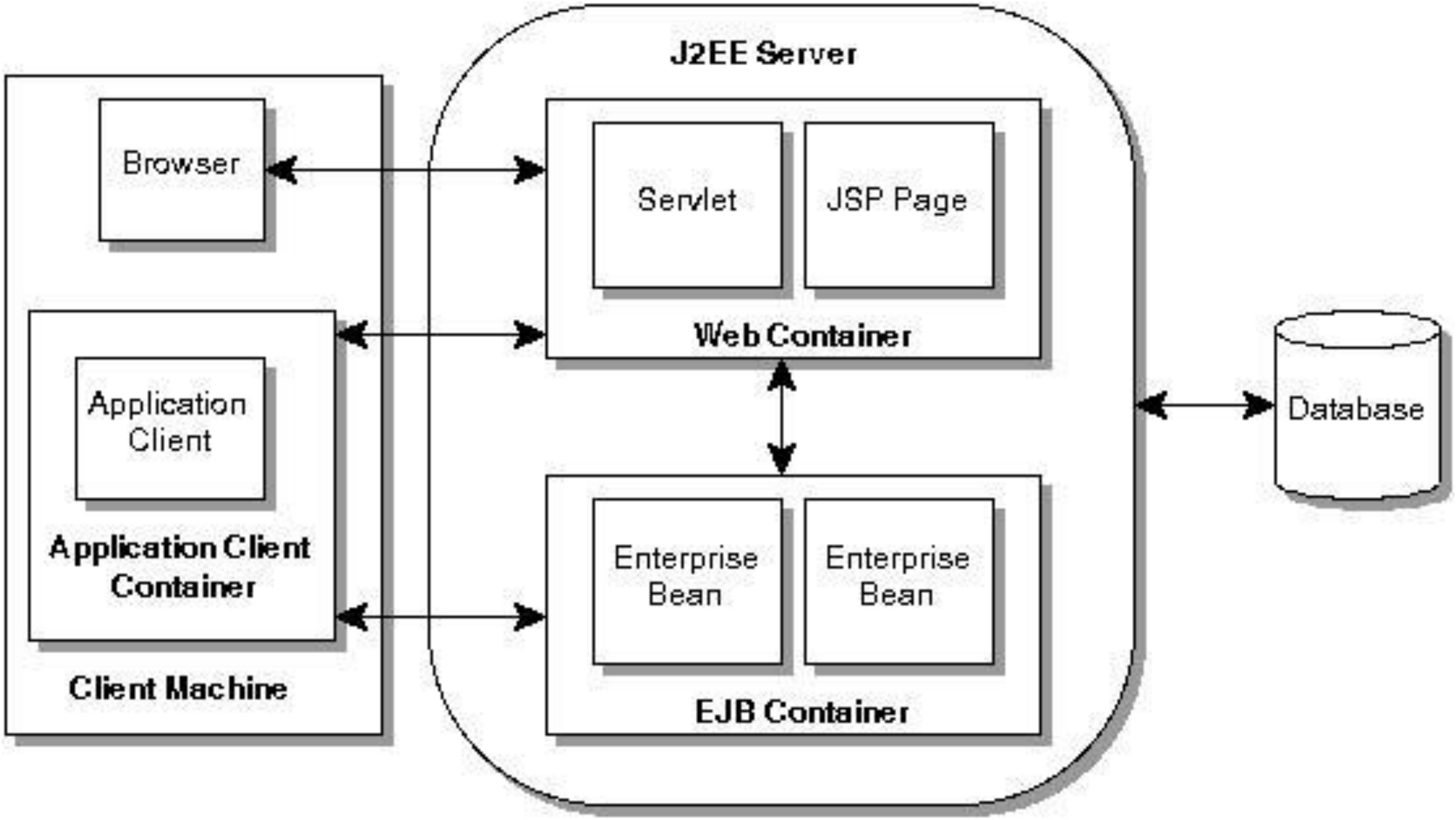
# Disadvantages of Existing System

- Heavy work load because maintained in the form of files or records.
- Time consumption is high.
- Difficult to data maintenance and search.
- It taken more memory space.
- Data can be loss.
- To avoid all these limitation and make the working more accurately the system need to be computerized.

# Proposed System

- The main objective of the existing system is to provide a user-friendly interface.
- Due to inconvenience in managing the data using leave or on duty forms, it is required to have a computer based system where a student can login and apply for the leave.
- Once the details are fed into the computer there is no need for various persons to deal with separate sections.
- Only a single person is enough to maintain all the reports.

# Overall System Architecture



# Modules Classification

- ADMIN
- STUDENT
- FACULTY

# MODULE 1 : ADMIN

In Admin module, users logged into the system are capable of doing the following operations. He/she can change the details of the students and faculties.

- **Add new user**

  Admin can add new student or faculty by giving separate id and name as input. The default password on creation is “KSRCE” for all created new staff. Here, the new student and faculty is added. The admin only have rights to do this for a security purpose.

- **Update existing user**

  Admin can update existing student or faculty details by giving their id. Using this module admin can update the basic details. Admin can also change the user role of faculty.

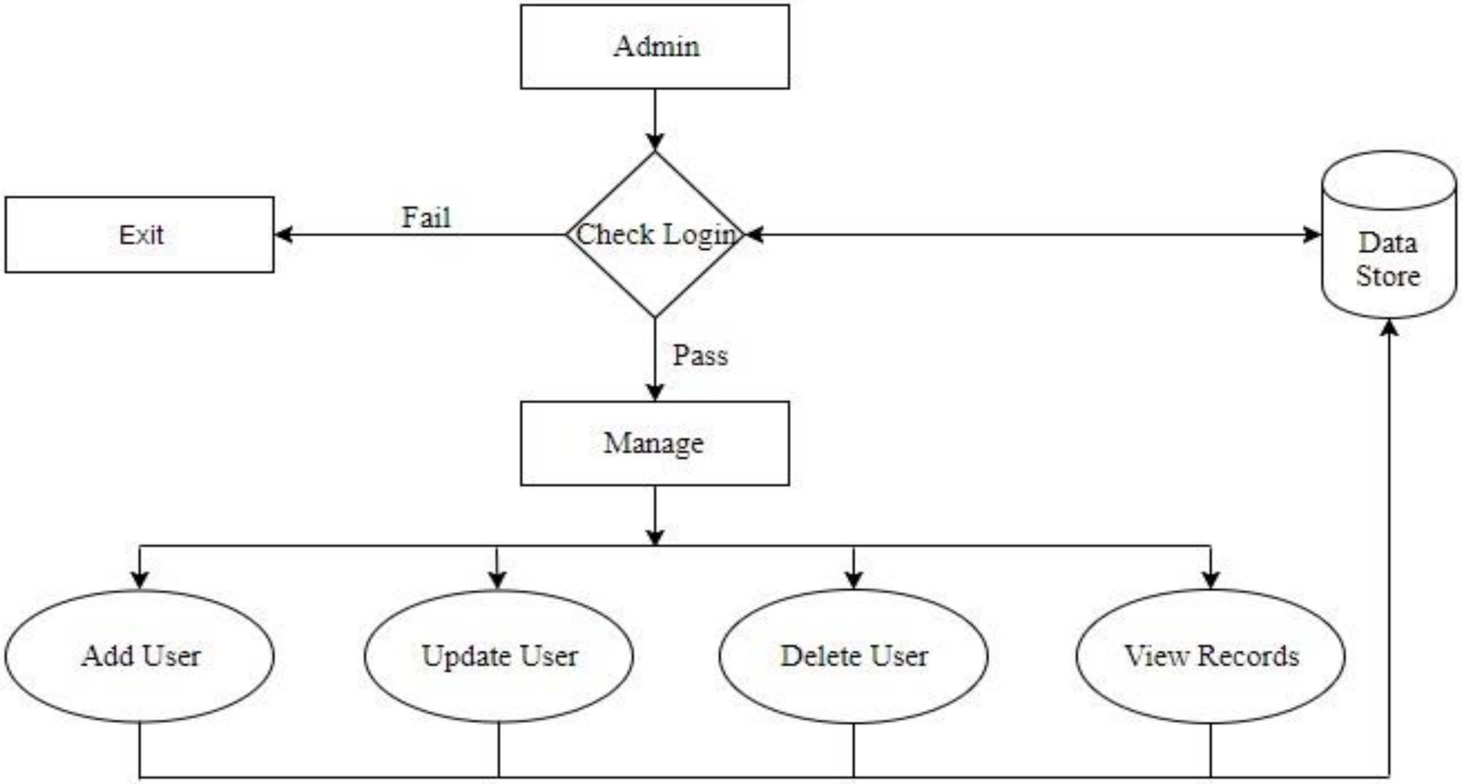
- **Delete existing user**

  Admin can delete student or faculty name or details from the system if they are not need. All details of student or faculty are deleted permanently from the system. i.e., if anyone has relieved their details will be erased.

- **Change Password**

  Admin can change login password using this module. So that if the password is forgot by the student or faculty can be changed by admin.

# MODULE 1  Diagram



# MODULE 1 Coding with explanation

- To insert a record into the student table,

  ```
  INSERT INTO student (studentid, studentname, regno, year, section, password)
  VALUES(1, 'raj', '1413075', 4, 'c', '1234');
  ```

- To update details in the faculty table,

  ```
  UPDATE faculty SET password = '5678' WHERE username='raj';
  ```

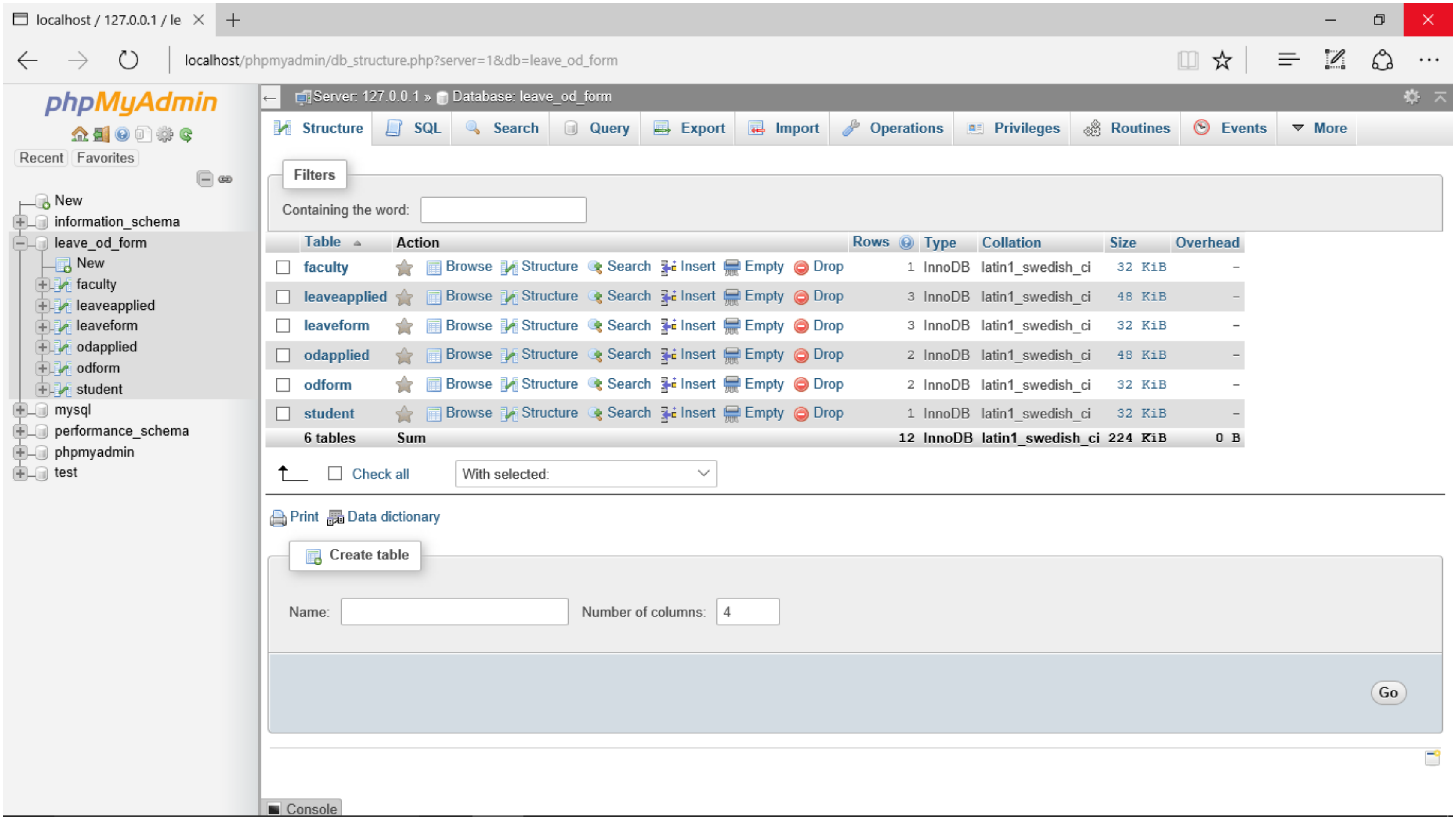
- To delete a record in the student table,

  ```
  DELETE FROM student WHERE regno = '1413075';
  ```

- To view particular records,

  ```
  SELECT * FROM student WHERE regno = '1413075';
  ```

# MODULE 1 Snapshot



localhost/phpmyadmin/db_structure.php?server=1&db=leave_od_form

Server: 127.0.0.1 » Database: leave_od_form

Filters

Containing the word:

| | Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|---|
| □ | faculty | ☆ Browse Structure Search Insert Empty Drop | 1 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| □ | leaveapplied | ☆ Browse Structure Search Insert Empty Drop | 3 | InnoDB | latin1_swedish_ci | 48 KiB | - |
| □ | leaveform | ☆ Browse Structure Search Insert Empty Drop | 3 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| □ | odapplied | ☆ Browse Structure Search Insert Empty Drop | 2 | InnoDB | latin1_swedish_ci | 48 KiB | - |
| □ | odform | ☆ Browse Structure Search Insert Empty Drop | 2 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| □ | student | ☆ Browse Structure Search Insert Empty Drop | 1 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| | 6 tables | Sum | 12 | InnoDB | latin1_swedish_ci | 224 KiB | 0 B |

□ Check all With selected:

Create table

Name: Number of columns: 4

# MODULE 2 : STUDENT

In Student module, users logged into the system are capable of doing the following operations. Here the student can apply leave or on duty and can view the details of the applied leave and on duty.

- **6.3.1 Apply leave or on duty**

  Students can apply leave and on duty through designed forms by giving the proper details.

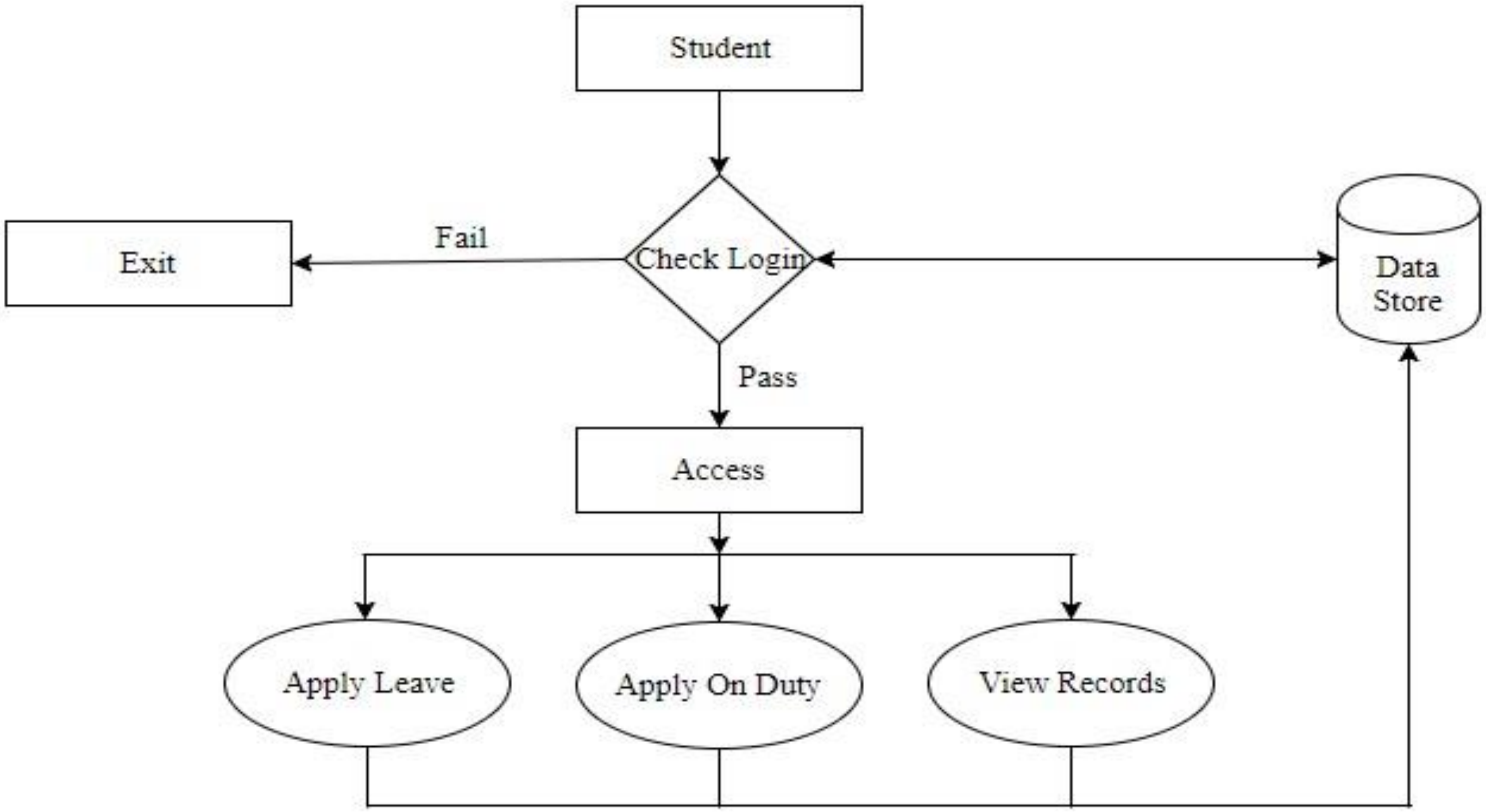
- **6.3.2 View record**

  Students can view the applied leave and on duty details. This is also used know whether the applied leave or on duty is accepted or rejected by the faculty.

- **6.3.3 Logout**

  Student should logout securely from the system.

# MODULE 2 Diagram

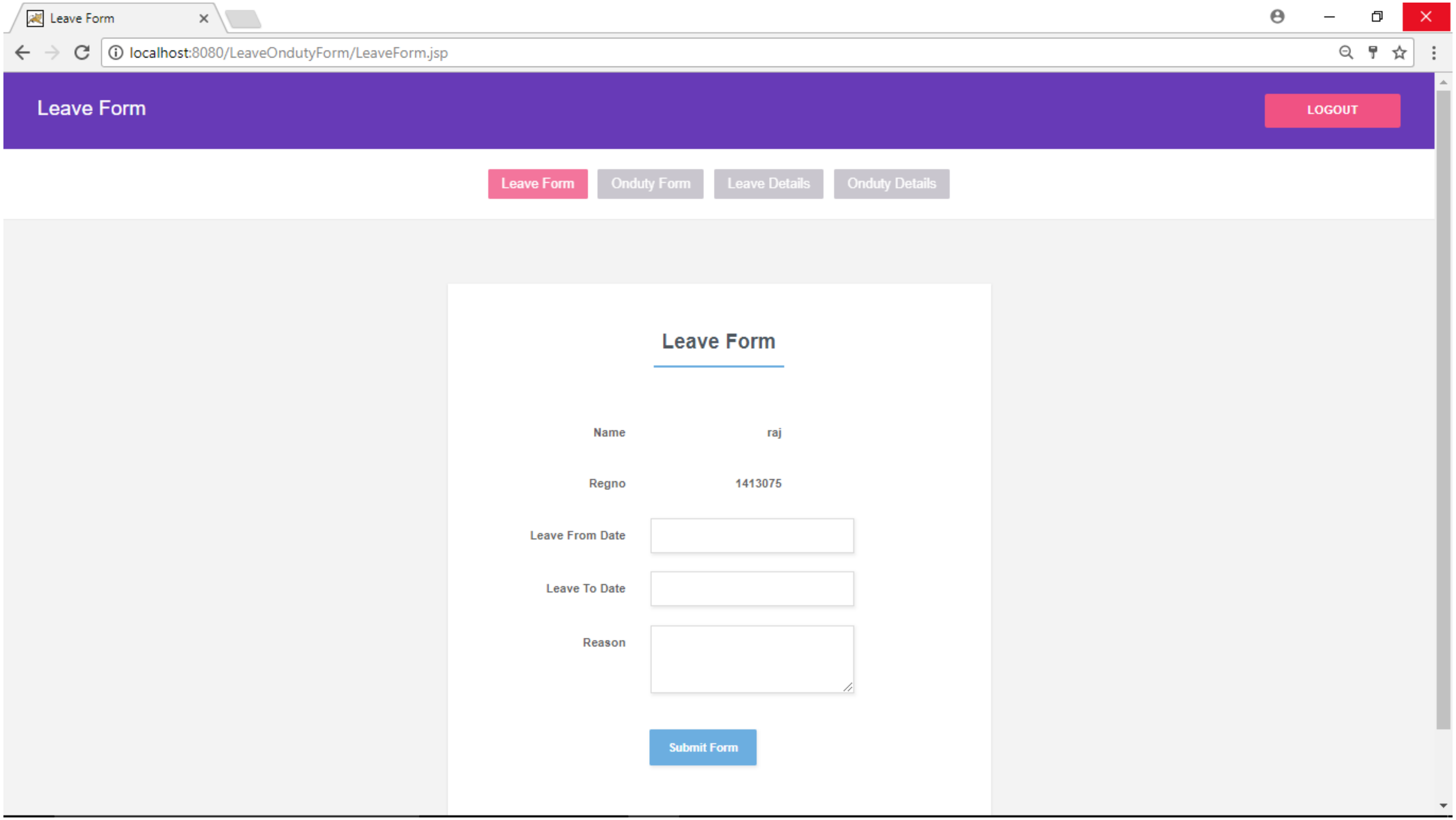


# MODULE 2 Coding with explanation

- To apply leave from student login,

```
public boolean applyLeave(String receivedFromDate, String receivedToDate,String
                                   receivedReason,String receivedRegno)
{
    preparedStatement = con.prepareStatement("insert into leaveform
       (studentid,dateofapply,dateofleavefrom,dateofleaveto,reason) values
              (select studentid from student where regno=?), now(),?,?,?)”);
    preparedStatement.setString(1,receivedRegno);
    preparedStatement.setString(2,receivedFromDate);
    preparedStatement.setString(3,receivedToDate);
    preparedStatement.setString(4, receivedReason);
    update1 = preparedStatement.executeUpdate();
}
```

# MODULE 2 Snapshot



| | |
|---|---|
| Name | raj |
| Regno | 1413075 |
| Leave From Date | |
| Leave To Date | |
| Reason | |

# MODULE 3 : FACULTY

In Faculty module, users logged into the system are capable of doing the following operations. Here the faculty is responsible for approve and view the details of the leave and on duty applied by the students.

- **6.2.1 Approve record**

  Faculty can approve the records of leave and on duty applied by the student by giving separate id or name as input. The approval can be accept or reject.

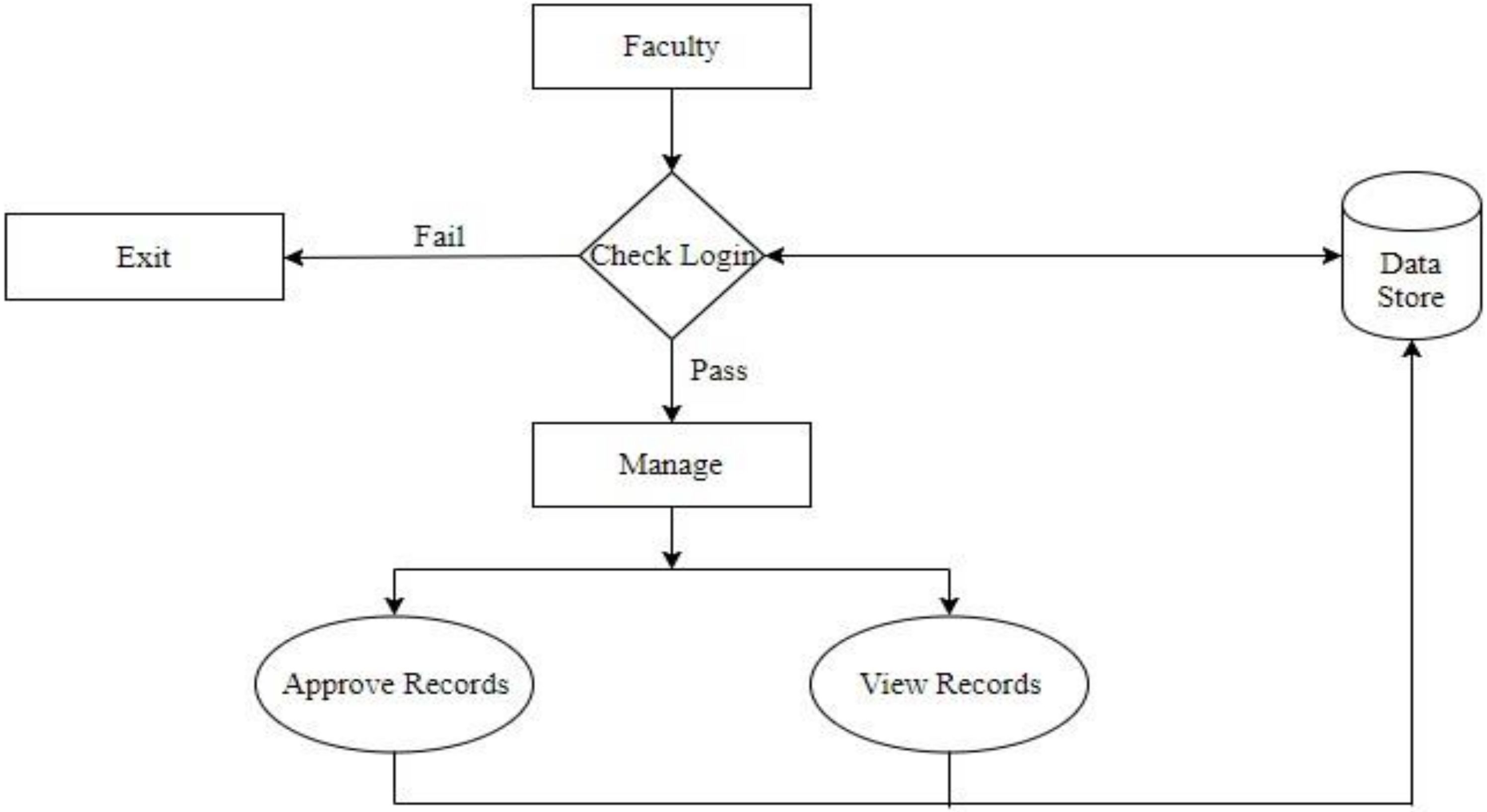
- **6.2.2 View record**

  Faculty can view all records of leave and on duty applied by the student by giving separate id or name as input or the records can be searched through other user details.

- **6.2.3 Logout**

  Faculty should logout securely from the system.

# MODULE 3 Diagram



# MODULE 3 Coding with explanation

- To view the applied leave records,

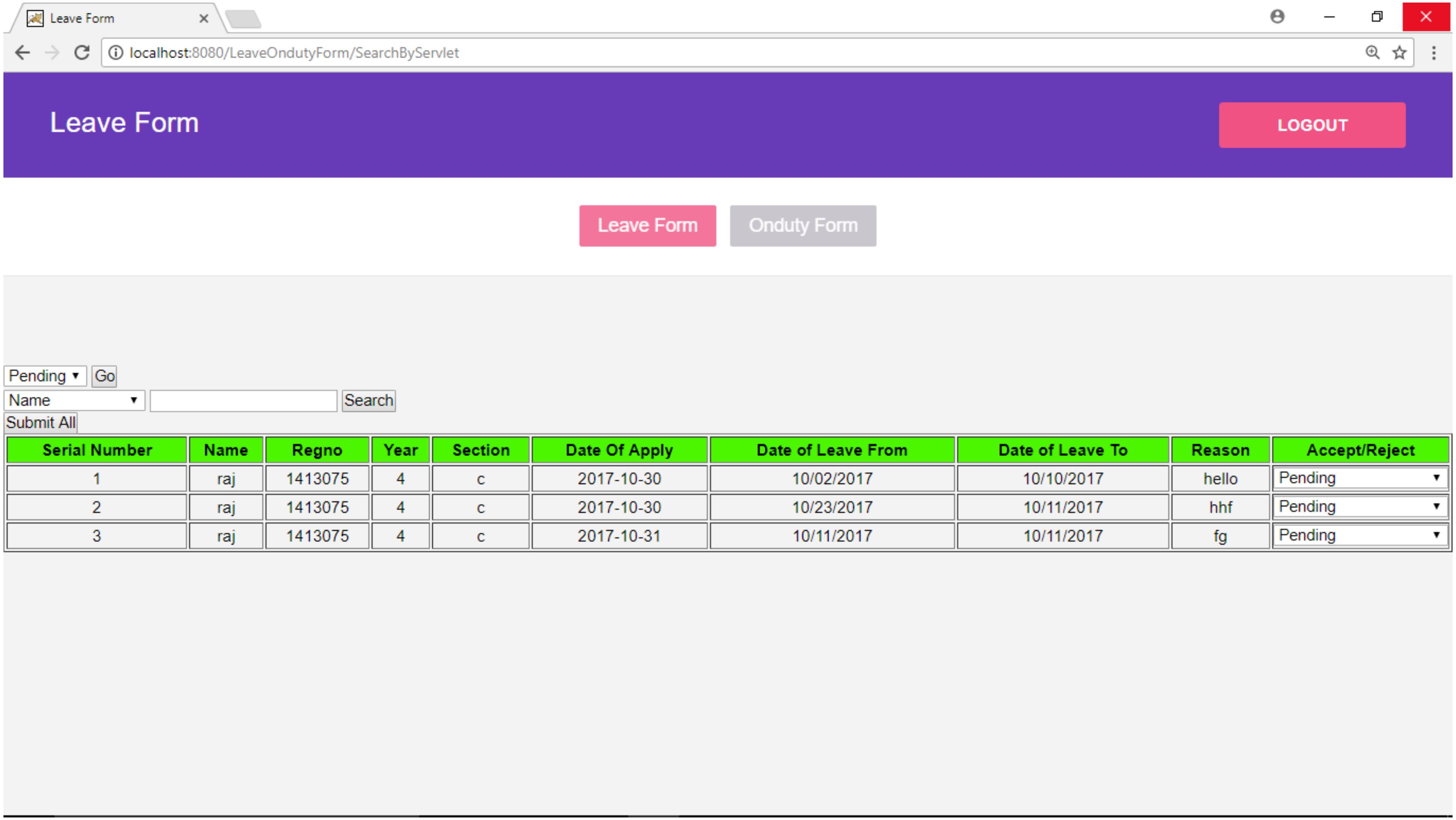
```
<%
    LeaveAppliedFacultyDao obj= new LeaveAppliedFacultyDao();
    LinkedList<LeaveFormBean> leaveFormList=obj.leaveApplied("pending");
    Iterator<LeaveFormBean> listIterator = leaveFormList.iterator();

    while(listIterator.hasNext())
    {
            LeaveFormBean  leaveFormBeanObj = listIterator.next();
 %>
    <td> <% out.print(leaveFormBeanObj.getDateOfApply()); %></td>
    <td> <% out.print(leaveFormBeanObj.getDateOfLeaveFrom()); %></td>
    <td> <% out.print(leaveFormBeanObj.getDateOfLeaveTo());%> </td>
    <td> <% out.print(leaveFormBeanObj.getReason()); %></td>
```

# MODULE 3 Coding with explanation

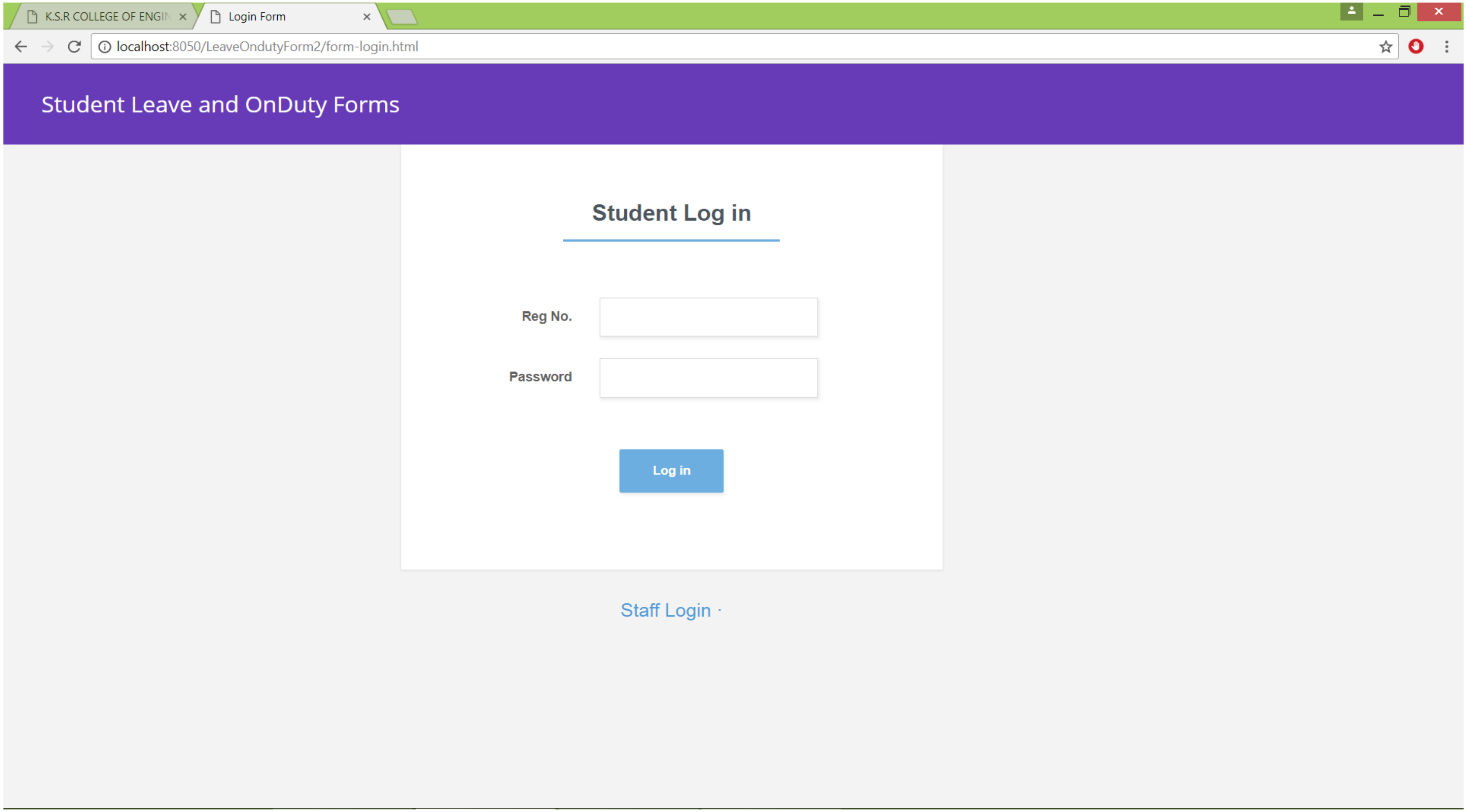
```
<td>
    <select name="facultyselection" style="width:100%;">
            <option value="pending" selected="selected">Pending</option>
            <option value="accept">Accept</option>
            <option value="reject">Reject</option>
    </select>
</td>
 <% } %>
```

# MODULE 3  Snapshot



| Serial Number | Name | Regno | Year | Section | Date Of Apply | Date of Leave From | Date of Leave To | Reason | Accept/Reject |
|---|---|---|---|---|---|---|---|---|---|
| 1 | raj | 1413075 | 4 | c | 2017-10-30 | 10/02/2017 | 10/10/2017 | hello | Pending ▾ |
| 2 | raj | 1413075 | 4 | c | 2017-10-30 | 10/23/2017 | 10/11/2017 | hhf | Pending ▾ |
| 3 | raj | 1413075 | 4 | c | 2017-10-31 | 10/11/2017 | 10/11/2017 | fg | Pending ▾ |

# Overall Snapshot



K.S.R COLLEGE OF ENGIN

Login Form

localhost:8050/LeaveOndutyForm2/form-login.html

Student Leave and OnDuty Forms

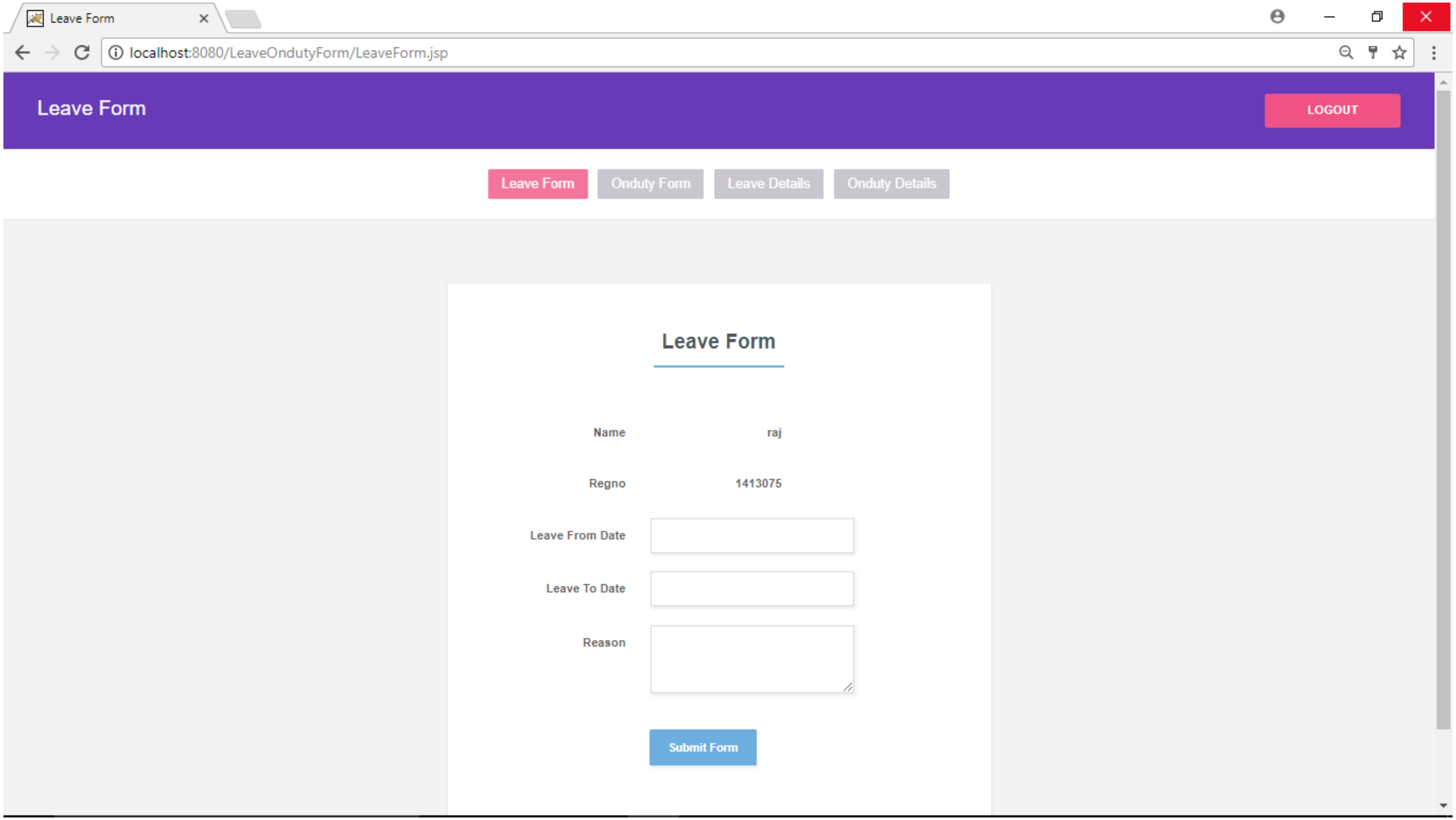
Student Log in

Reg No.

Password

Staff Login ·

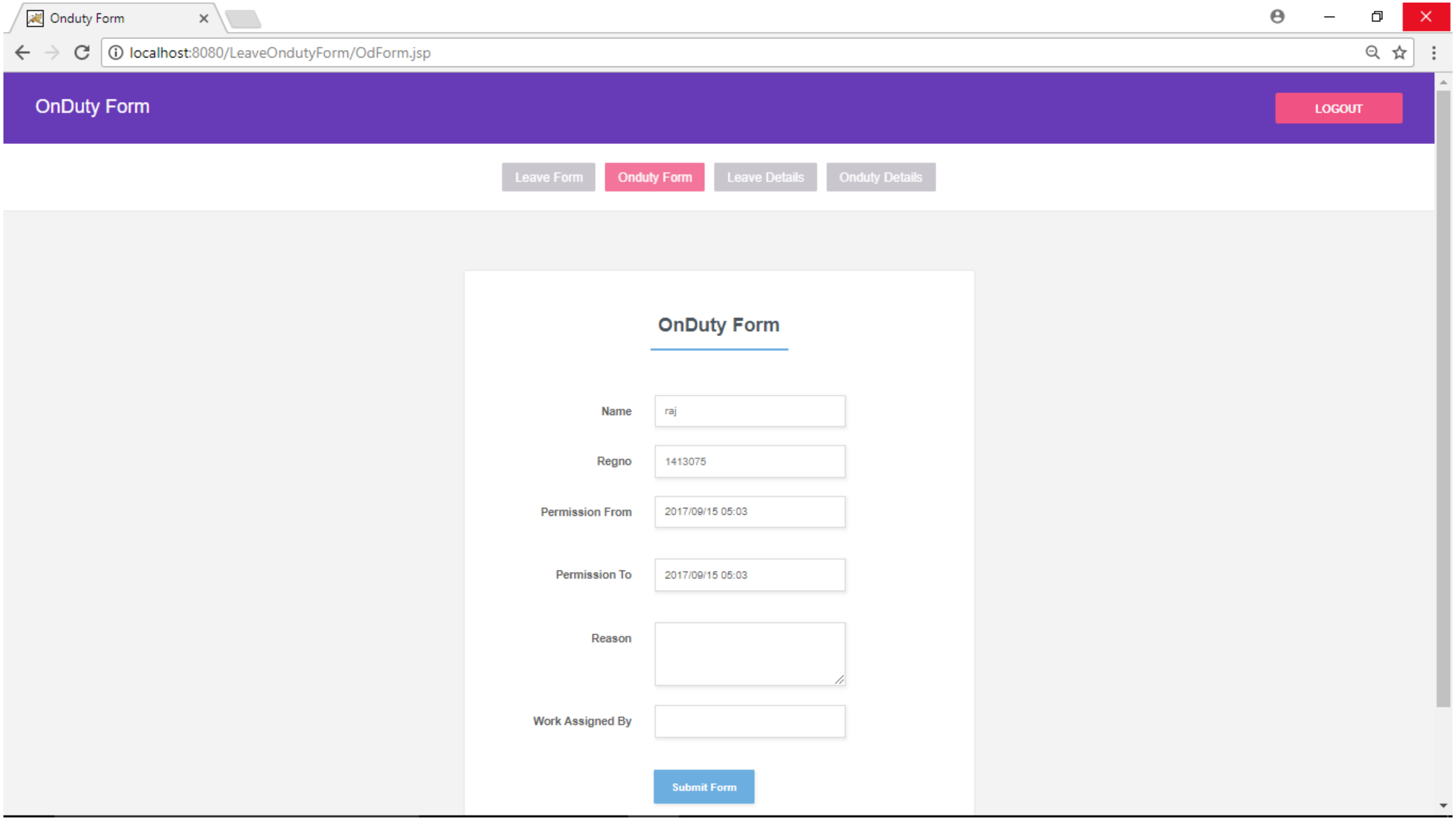
# Overall Snapshot



## Leave Form

| | |
|---|---|
| Name | raj |
| Regno | 1413075 |
| Leave From Date | |
| Leave To Date | |
| Reason | |

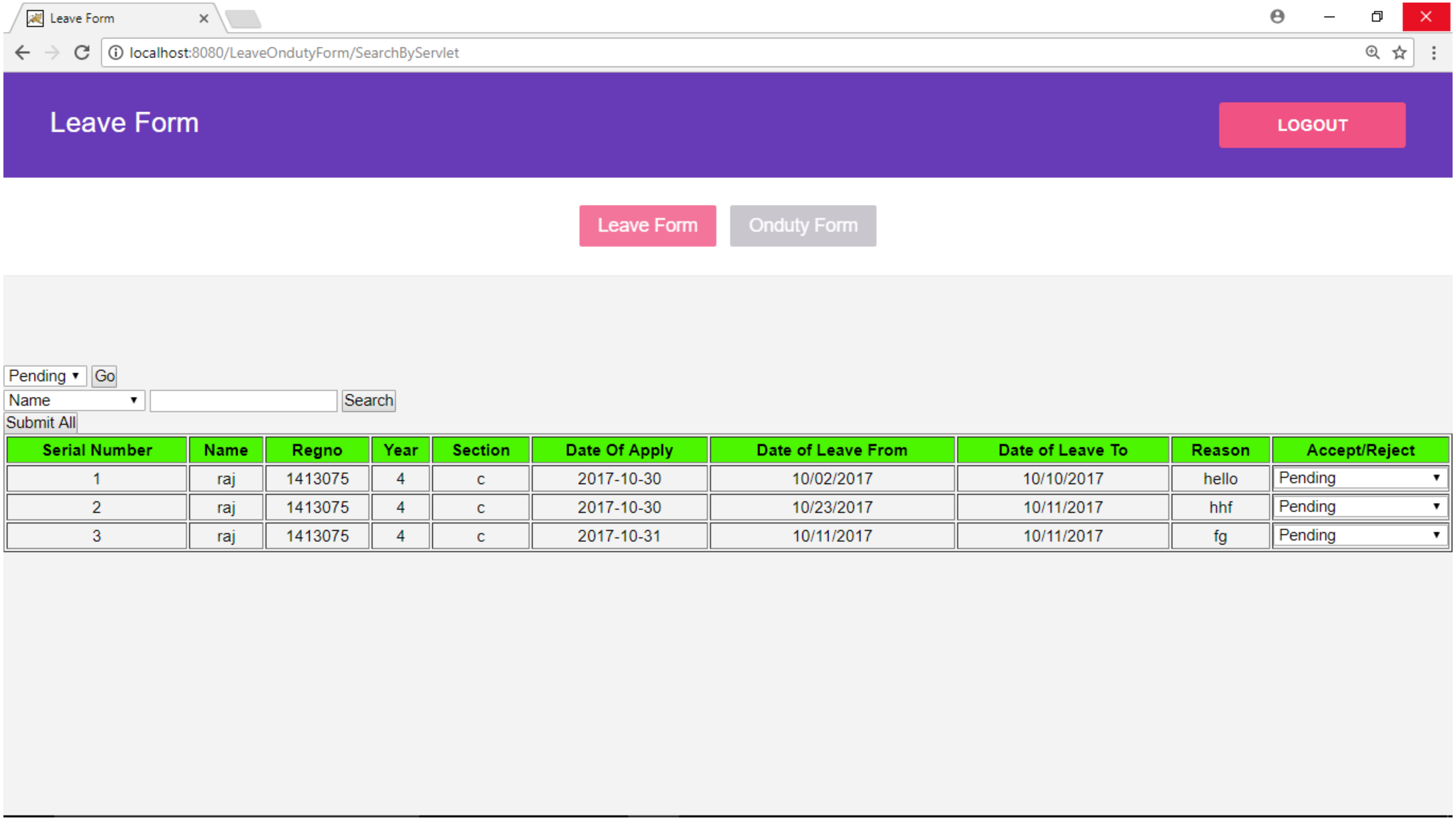
# Overall Snapshot



OnDuty Form

|  |  |
|---|---|
| Name | raj |
| Regno | 1413075 |
| Permission From | 2017/09/15 05:03 |
| Permission To | 2017/09/15 05:03 |
| Reason |  |
| Work Assigned By |  |

# Overall Snapshot



| Serial Number | Name | Regno | Year | Section | Date Of Apply | Date of Leave From | Date of Leave To | Reason | Accept/Reject |
|---|---|---|---|---|---|---|---|---|---|
| 1 | raj | 1413075 | 4 | c | 2017-10-30 | 10/02/2017 | 10/10/2017 | hello | Pending ▾ |
| 2 | raj | 1413075 | 4 | c | 2017-10-30 | 10/23/2017 | 10/11/2017 | hhf | Pending ▾ |
| 3 | raj | 1413075 | 4 | c | 2017-10-31 | 10/11/2017 | 10/11/2017 | fg | Pending ▾ |

# Project Demo

# References

- Front End : http://www.w3schools.com
- J2EE : http://www.javatpoint.com/java-tutorial
- Java : http://www.tutorialspoint.com/java
- JDBC: : https://www.javatpoint.com/java-jdbc
- Servlet : http://www.servlets.com
- Queries : http://www.stackoverflow.com
- Database : https://dev.mysql.com
- Book : Java - Complete Reference by Herbert Schildt