

**A**  
**MINI PROJECT REPORT**  
**ON**  
**RESOURCE ALLOCATION SYSTEM**

**SUBMITTED**

**BY**

**KAREDLA ANANTHA SASHI SEKHAR**  
**(1601-14-733-091)**  
**&**  
**M DASARADA RAM REDDY**  
**(1601-14-733-092)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY[A]**  
[Accredited by NBA–AICTE & NAAC-UGC, ISO 9001-2008 Certified & UGC Autonomous Institute]  
Website: [www.cbit.ac.in](http://www.cbit.ac.in), [www.cbitworld.com](http://www.cbitworld.com)  
HYDERABAD – 500 075  
2016-2017



**CHAITANYA BHARATHI INSTITUTE OF  
TECHNOLOGY[A]  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
GANDIPET, HYDERABAD-500 075**

**CERTIFICATE**

This is to certify that the project work entitled “**RESOURCE ALLOCATION SYSTEM**” submitted to **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY**, in partial fulfilment of the requirements for the award of the completion of Third year 2nd semester of B.E in Computer Science and engineering, during the academic year 2016-17, is a record of original work done by **K.A.S.SEKHAR (160114733091)** and **M.DASARADA RAM REDDY (160114733092)** during the period of study in the Dept. of CSE, CBIT, HYDERABAD, under our supervision and guidance.

**PROJECT GUIDE**

**G. Vivek**  
Assistant Prof,  
Dept of CSE.

**HEAD OF THE DEPARTMENT**

**Y Rama Devi**  
Head of the department,  
Computer Science,  
C.B.I.T, HYDERABAD

## ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Dr. B. Chennakeshava Rao, The Principal, C.B.I.T., Hyderabad** and **Y Rama Devi, Head of Department, Computer Science** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude to our guide **Mr G. Vivek, Assistant Professor, Dept of CSE, C.B.I.T.,** for his kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

**K.A.S.SEKHAR**  
**(1601-14-733-091)**

**&**

**M DASARADA RAM REDDY**  
**(1601-14-733-092)**

## **DECLARATION**

We declare that the project report entitled “**RESOURCE ALLOCATION SYSTEM**” is being submitted by us in the Department of computer science, Chaitanya Bharathi Institute of Technology.

This is record of bonafide work carried out by us under the guidance and supervision of **Mr G.Vivek, Assistant Professor, Dept of CSE, C.B.I.T.**

No part of the thesis copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The reported are based on the project work doing entirely by us and not copied from any other source.

**K.A.S.SEKHAR**  
**(1601-14-733-091)**  
**&**  
**M DASARADA RAM REDDY**  
**(1601-14-733-092)**

## **ABSTRACT**

This project is aimed at automating a Resource Allocation system. Every individual has to request for a resource by providing certain basic details of his/her requirements. After a successful registration, the user portal can be accessed with the username and the respective password. Seminar Halls, Assembly Halls, Labs etc can be requested from the available resources list.

The system allows a user to register and create an account by providing details and an authenticating password. The user can then log into to his/her account by entering the credentials. This is a user-friendly interface. Admin has the power to reject a particular request and to de-allocate a particular allocated entry during crisis. A user entity possesses all the entities like number of rejects so as to help admin to make a proper decision.

## TABLE OF CONTENTS

Acknowledgement	iii
Declaration	iv
Abstract	v

<b>Contents</b>	<b>Page No.</b>
1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Overview	1
2. Literature Survey	2
2.1 Java Servlets	2
2.1.1 Introduction	2
2.1.2 Java Servlets and Their Applications	2
2.1.3 Architecture of Servlets	2
2.2 Java Server Pages	3
2.3 JDBC Data Access	4
3. Software Requirements and Specifications	5
4. Design - Data Flow Diagram	6
5. Implementation	8
5.1 Module specification	8
5.1.1 Administrator Account	8
5.1.2 User Account	9
5.2 Working	9
6. Result	10
7. Conclusion	18
8. Bibliography	19

## LIST OF FIGURES

<b>Fig.No</b>	<b>FIGURE NAME</b>	<b>Page No.</b>
6.1	Output Screen of Home page for all users	10
6.2	Output Screen of Admin login	10
6.3	Output Screen of Admin page	11
6.4	Output Screen of Admin logging out	11
6.5	Output Screen after Admin logged out	12
6.6	Output Screen of Student registering	13
6.7	Output Screen on Successful registration of user	13
6.8	Output Screen of registration page username warning	14
6.9	Output Screen of password warning on registration page	14
6.10	Output Screen of user Logging in	15
6.11	Output Screen after user Logged in ( Booking page )	15
6.12	Output Screen after Successful booking	16
6.13	Output Screen of user Logging out	16
6.14	Output Screen after user Logged out	17

# 1. INTRODUCTION

## 1.1 Purpose:

Resources Present in college are usually very few and the requirements are very large in number. The main challenge for the person in authority is to allocate the requests reasonably and to make sure that no party is continuously rejected for some reason. Users on the other end have a hard time to make their requests manually by writing letters or any other form of written document. Hence in the system proposed a user friendly UI is designed where they can make requests easily, there will be proper means to store the previous history of allocations and there will means to make emergency requests.

## 1.2 Scope:

### *Functionality Scope:*

- Create a user account by accepting information.
- Login to account and make requests.
- Maintaining a list of available resources and requests.

### *Implementation Scope:*

The application is a web app so it is operation system independent. It is easier to run the software on any existing browser.

## 1.3 Overview:

This application enables users to create accounts with their details and access a list of available resources via those accounts. This makes reserving resources hassle-free and encourages students to use the college resources more often. Requests are added to overall requests table which is monitored by the admin , hence forth admin can make the necessary allotments.



## 2. Literature Survey

### 2.1. Java Servlets

#### 2.1.1. Introduction

Servlet can be described in many ways, depending on the context.

- Servlet is a technology i.e. used to create web application.
- Servlet is an API that provides many interfaces and classes including documentations.
- Servlet is an interface that must be implemented for creating any servlet.
- Servlet is a class that extends the capabilities of the servers and respond to the incoming request. It can respond to any type of requests.
- Servlet is a web component that is deployed on the server to create dynamic web page.

#### 2.1.2. Java Servlets and Their Applications

There are many advantages of servlet over CGI. The web container creates threads for handling the multiple requests to the servlet. Threads have a lot of benefits over the Processes such as they share a common memory area, lightweight, cost of communication between the threads are low. The basic benefits of servlet are as follows:

1. **Better performance:** because it creates a thread for each request not process.
2. **Portability:** because it uses java language.
3. **Robust:** Servlets are managed by JVM so we don't need to worry about memory leak, garbage collection etc.
4. **Secure:** because it uses java language.

#### 2.1.3. Architecture of Servlets

**Servlet** technology is used to create web application (resides at server side and generates dynamic web page). **Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was popular as a server-side programming language. But there were many disadvantages of this technology. We have discussed these disadvantages below. There are many interfaces and classes in the servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

## 2.2 Java Server Pages

**JSP** technology is used to create web application just like Servlet technology. It can be thought of as an extension to servlet because it provides more functionality than servlet such as expression language, jstl, etc. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tag etc.

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified.

JSP can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other Java program, they must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well.

The Web container creates JSP implicit objects like pageContext, servletContext, session, request & response.

## 2.3 JDBC Data Access

Java JDBC is a java API to connect and execute query with the database. JDBC API uses JDBC drivers to connect with the database. Before JDBC, ODBC API was the database API to connect and execute query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

### 3. SOFTWARE REQUIREMENT SPECIFICATIONS

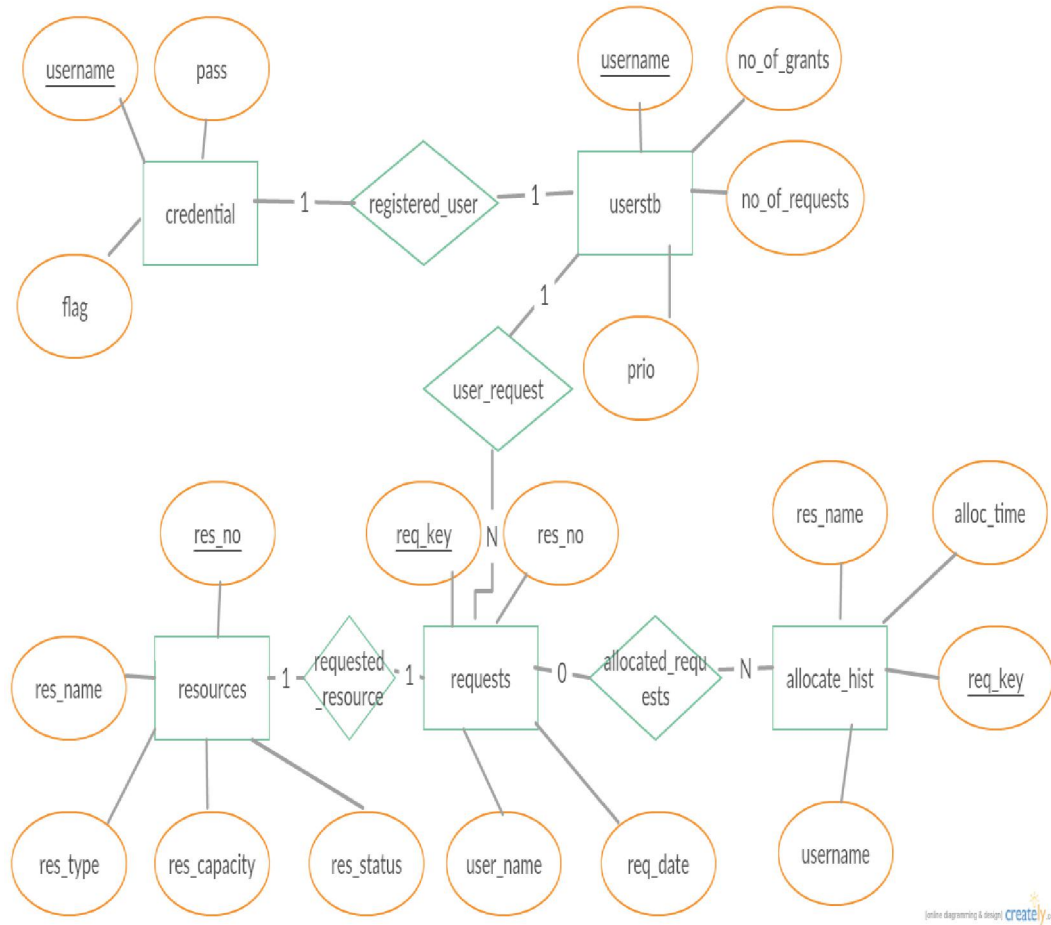
- **Software Requirements:**

- Operating System : Windows XP,7,8, Linux
- User Interface : Sublime Text Editor
- Programming Language : HTML,JSP,JAVA

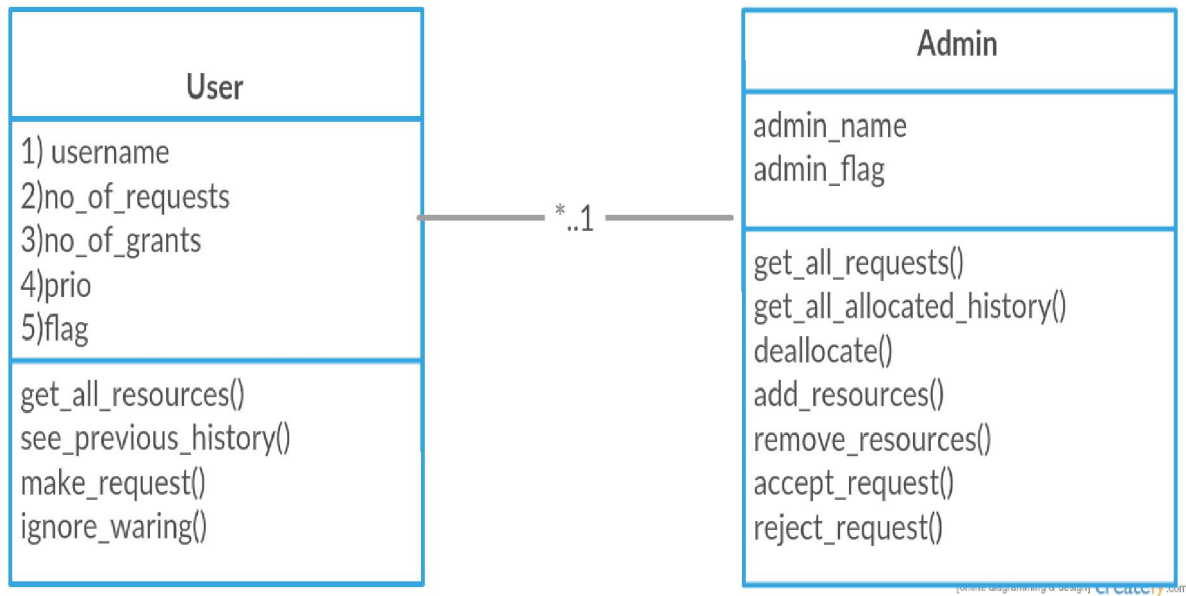
- **Minimum Hardware Requirements:**

- Processor : 1.6 GHz processor
- RAM : 1 GB
- Operating System : Windows XP,7,8
- Disk Space : 1 GB

## 4.DESIGN (ER DIAGRAM)



## (UML DIAGRAM)



## **5. IMPLEMENTATION**

### **5.1 Module specification:**

The following web app is divided into two modules:

- Administrator: The administrator has the privileges to view the list of requests, registered users and the resources.
- User: This user has the privilege to request a resource after registering and logging in using username and password.

#### **5.1.1 Home Page:**

This page is common to both the administrator and user. This page has the options to login and register. Based on the login it is decided whether it is a user login or admin login.

#### **5.1.2 Administrator Account**

Administrator of this application has the privilege of viewing requests and all the data available in the database.

#### **Admin Page:**

After logging in using the correct username and password, administrator will be redirected to a page where he/she will be able to view all the data in the database like requests, allocate\_hist, userstb. This page can be seen in fig.

The admin will be able to see all the previous requests and their status(accepted or rejected). To the admin all the requests will appear as a list, the admin will have two options, either to reject the request or to allocate the request. Clicking on either button will take the admin to the corresponding message displaying page.

#### **Logout:-**

Pressing logout button terminates the session and takes the admin to the HOME PAGE.

### **5.1.3 User Account**

User has the privilege to request a resource before hand by choosing a resource from the list of resources displayed.

#### **Register:**

To book a resource the user or student should be registered. To register user has to click on the register link on the home and enter all the details in the redirected page and click on register to register. This page can be seen in fig.

#### **Login:**

To login, the user should be registered before. After entering correct username and password, the user will be redirected to a page where a list of resources is displayed. Here the user can request the resource he/she needs by pressing the allocate button adjacent to request. On clicking that button, user is redirected to another page where message is displayed. This page and login page are shown in figs.

#### **Logout:**

On clicking on logout, the user will be redirected to the home page. The session gets terminated.

#### **Warning:-**

Warning page is shown to user if the requested resource has been already allocated on the same date. The user is further given two options, one is not to make a request, other is to make a request anyway , the user hence makes a urgent request. The admin has the ability to deallocate the resource for a user and allocate it for another user.

## **5.2 Working**

The working of this web application involves fetching data from the database and comparing it with the entered data, updating the tables in database and performing different operations on it (CRUD operations). This is done with the help of using server side languages such as jsp and servlets which communicate with the database using JDBC connectivity.

We are also using session management to prevent unauthorized access of data and to maintain the session of the logged in user throughout the time.

## 6. RESULT

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/pro/'. The page title is 'Chaitanya Bharathi Institute of Technology'. Below the title is the institute's logo, which features a circular emblem with a book, a lamp, and the text 'CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY' and '1979'. The main heading is 'Resource Management System'. There are two input fields: 'Username' with placeholder text 'Enter Username' and 'Password' with placeholder text 'Enter Password'. Below these fields is a green 'Login' button. Under the 'Login' button is a checkbox labeled 'Remember me' and a link labeled 'register'.

Fig 6.1 Home page for all users

The screenshot shows the same web browser window as Fig 6.1, but with the 'Username' field containing the text 'admin' and the 'Password' field containing six dots. The 'Login' button is still green. The 'Remember me' checkbox is checked, and the 'register' link is still present.

Fig 6.2 Admin Login page



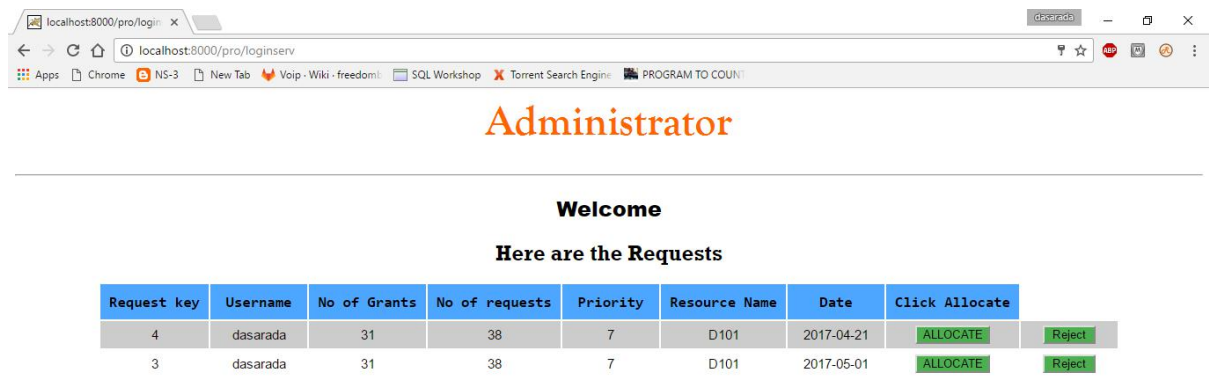
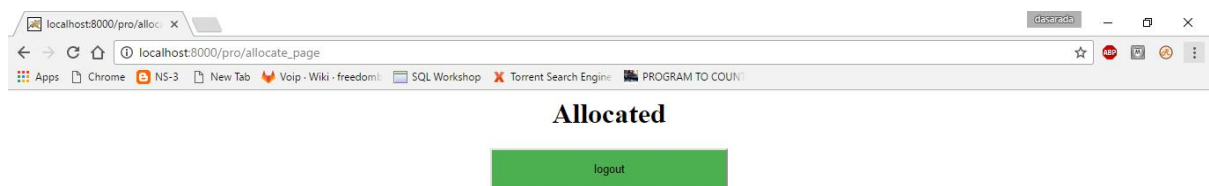


Fig 6.3 Admin page



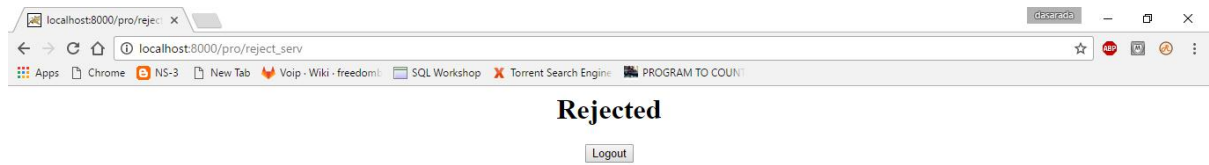


Fig 6.4 Admin logging out

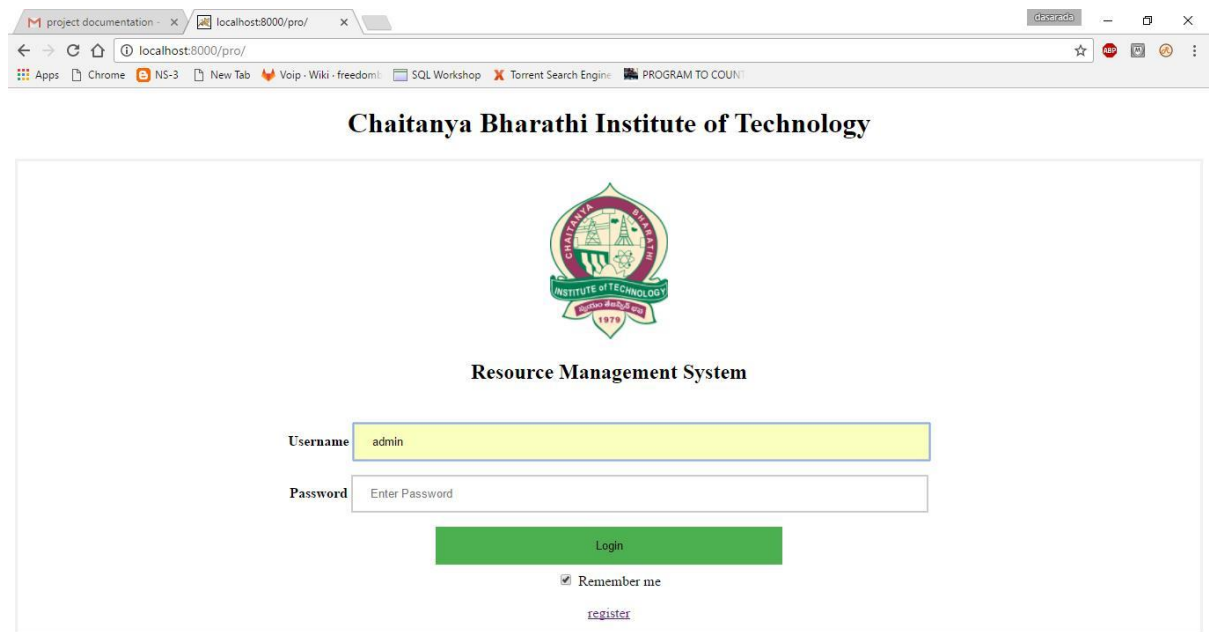


Fig 6.5 Admin redirected to home page after logging out

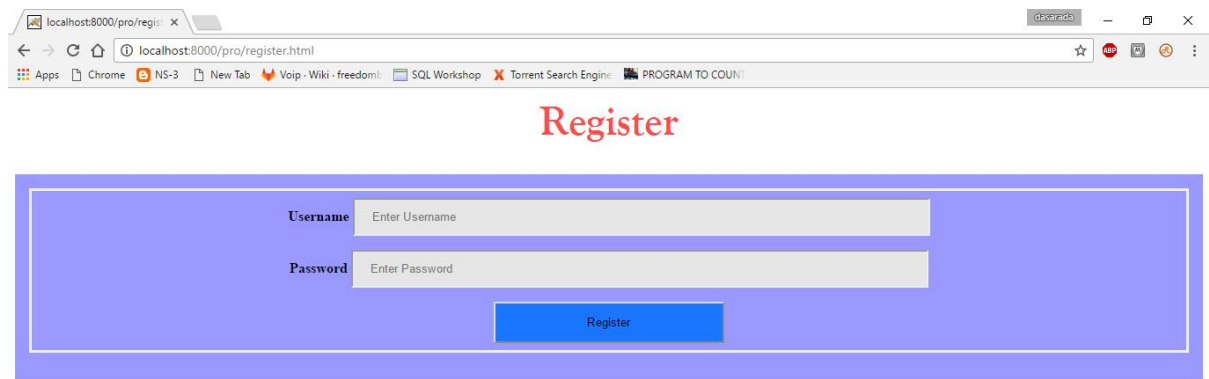


Fig 6.6 User Registration Page

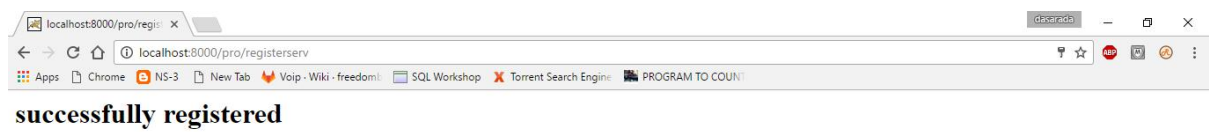


Fig 6.7 Successful User registration page

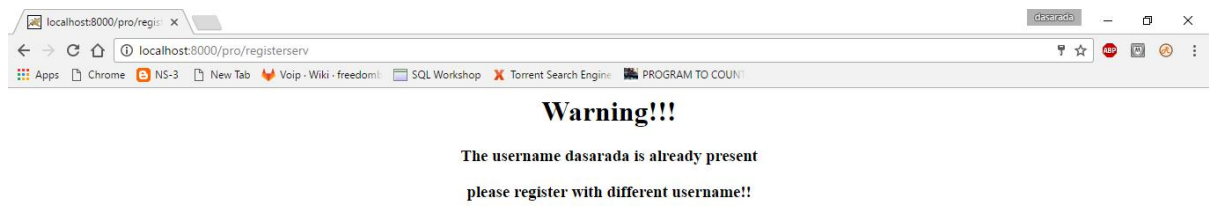


Fig 6.8 Register Page username warning

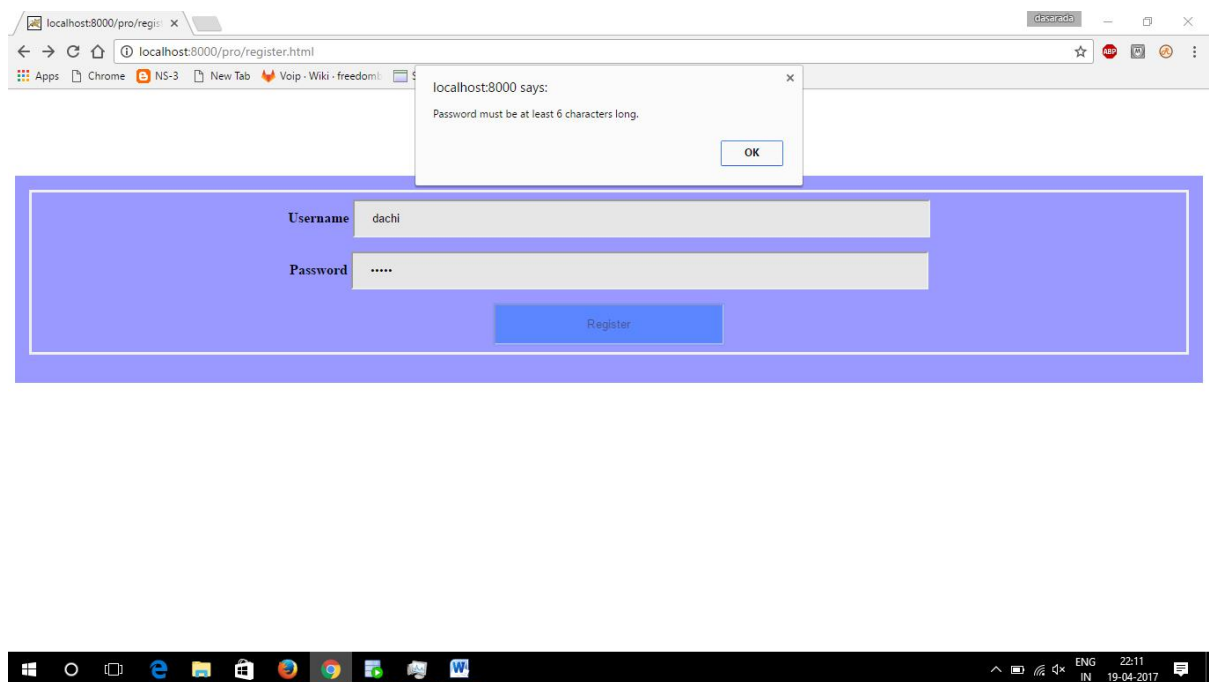



Fig 6.9 Register page Password warning

**Chaitanya Bharathi Institute of Technology**



**Resource Management System**

Username:

Password:

☒ Remember me

[register](#)

Fig 6.10 User Login page

**dasarada**

---

**Welcome**

Resource Number	Resource Name	Resource Type	Resource Capacity	Resource Status	Required Date	
1	D101	Seminar Hall	70	1	<input type="text" value="dd-mm-yyyy"/>	<input type="button" value="REQUEST"/>
2	Assembly hall	large seminar hall	250	1	<input type="text" value="dd-mm-yyyy"/>	<input type="button" value="REQUEST"/>
3	Mech seminar hall	seminar hall	120	1	<input type="text" value="dd-mm-yyyy"/>	<input type="button" value="REQUEST"/>

**History**

S No	Resource Name	Allocating time	Status
1	D101	2017-04-06	allocated
3	D101	2017-04-08	allocated
4	D101	2017-04-06	rejected
7	D101	2017-04-08	rejected
16	D101	2017-05-01	allocated
18	D101	2017-04-21	allocated
20	D101	2017-05-01	rejected

Fig 6.11 Requesting page

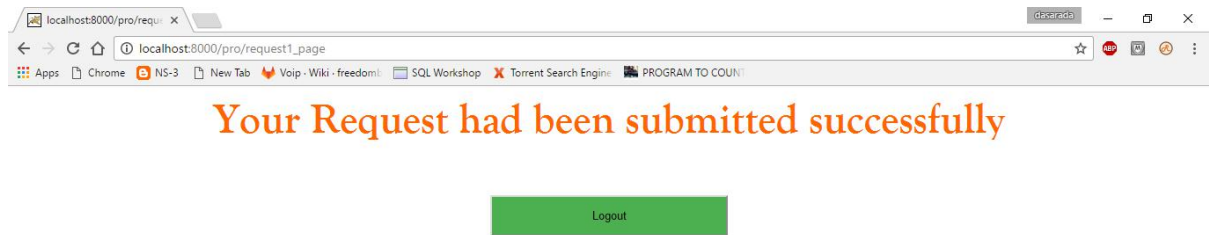


Fig 6.12 On successful booking

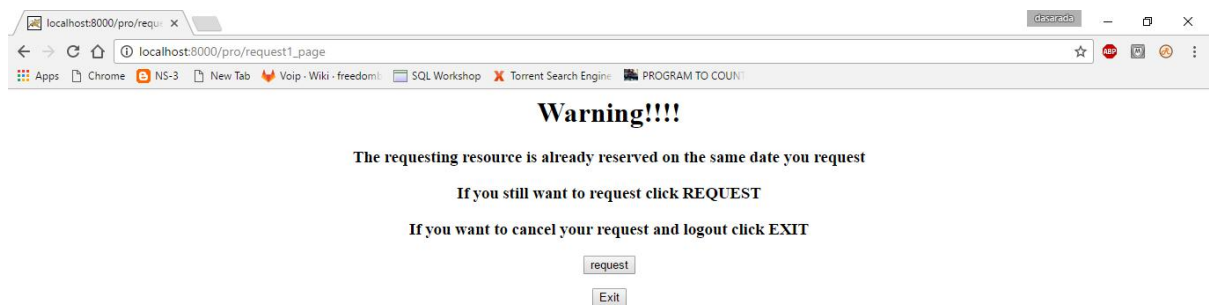



Fig 6.13 User Request warning page

project documentation - x localhost:8000/pro/ x

localhost:8000/pro/

Apps Chrome NS-3 New Tab Voip - Wiki - freedoml SQL Workshop Torrent Search Engine PROGRAM TO COUNT

## Chaitanya Bharathi Institute of Technology



### Resource Management System

**Username**

**Password**

☒ Remember me

[register](#)

Fig 6.14 User redirected to home page after logging out.

## **7. CONCLUSION**

Resource Management System is to automate the process of requesting and allocation of the available resources in an educational institution like Seminar Halls, Assembly Halls, Computer Labs, Canteen, Class rooms for various purpose such as Workshops, boot camps, guest lectures etc. This Automation is to reduce the hectic work of taking permissions of head of departments or concerned officials for various resources mentioned above, carrying request letters. It would save time of students and also avoids missing of classes for this work.



## 8. BIBILOGRAPHY

1. Tutorials Point.  
<http://www.tutorialspoint.com/>
2. W3 Schools.  
<http://www.w3schools.com/>
3. Code Academy.  
<https://www.codecademy.com/>
4. Java Point.  
<http://www.javapoint.com/>
5. Wikipedia  
<https://en.wikipedia.org/>
6. Stack Overflow.  
<https://www.stackoverflow.com/>
7. DBOracle.  
<http://www.dbaoracle.com/>