

EduBridge



A Project

Report

On

Placement Management System

By

Navya R

B.E

Batch: 2020 – 5883

Center: Bangalore

Under the Guidance of,

Mamta Boga.

Technical Trainer

Edu Bridge

(School of coding)

Introduction:

Our project explains about the Placement Management System by name Alchemy. This project mainly explains the various actions related to placement systems online. You can search jobs available in student module and add jobs according to category. Our Project include **Modules:**

- Admin Login
- Student Registration
- Student Login

I have developed this Application in **Java, JSP, Servlets, Hibernate and MySQL**. It's a web-based projects so I have used **HTML, CSS, JavaScript and Bootstrap also**.

The main feature of the project is to make placement management system easier. You can access this website from anywhere and everywhere.

In Admin module, admin can add the Jobs in the Add job category. He can also view the total number of students registered at student registration module in the website. Admin can only see the admin page, can add modify delete jobs and can see number of students enrolled.

In Student login, user can access the website and view the total number of jobs available in the cart. New user can register and login after that to registartion. In Cart, admin can access, they can add delete and modify the jobs if he want.

Software Requirements:

Front end: Java/J2EE technologies (Servlet, JSP), HTML, CSS, JavaScript, Bootstrap, Hibernate.

Back end: MySQL workbench 8.0.23CE.

Middleware/Server: Apache Tomcat v8.5. IDE: Eclipse IDE for Java EE Developers

Browser: Best result on Google Chrome

Data Dictionary:

- Create Database corejavadb;
- Hibernate: create table users(id INT NOT NULL AUTO_INCREMENT,contact_number VARCHAR(100) NOT NULL,name VARCHAR(100) NOT NULL,email VARCHAR(100) NOT NULL,password VARCHAR(100) NOT NULL,is_admin VARCHAR(20) NOT NULL,Date_Of_Birth VARCHAR(20) NOT NULL,user_name VARCHAR(20) NOT NULL,PRIMARY KEY (id));
- Hibernate: create table create table Drive(id INT NOT NULL AUTO_INCREMENT,company_name VARCHAR(100) NOT NULL,drive_date VARCHAR(100) NOT NULL,location VARCHAR(100) NOT NULL,PRIMARY KEY (id));

Screenshots:-

Home Page:

Welcome for Placement system

[Admin Login](#)

[Student Registration](#)

[Student Login](#)



Admin Login:

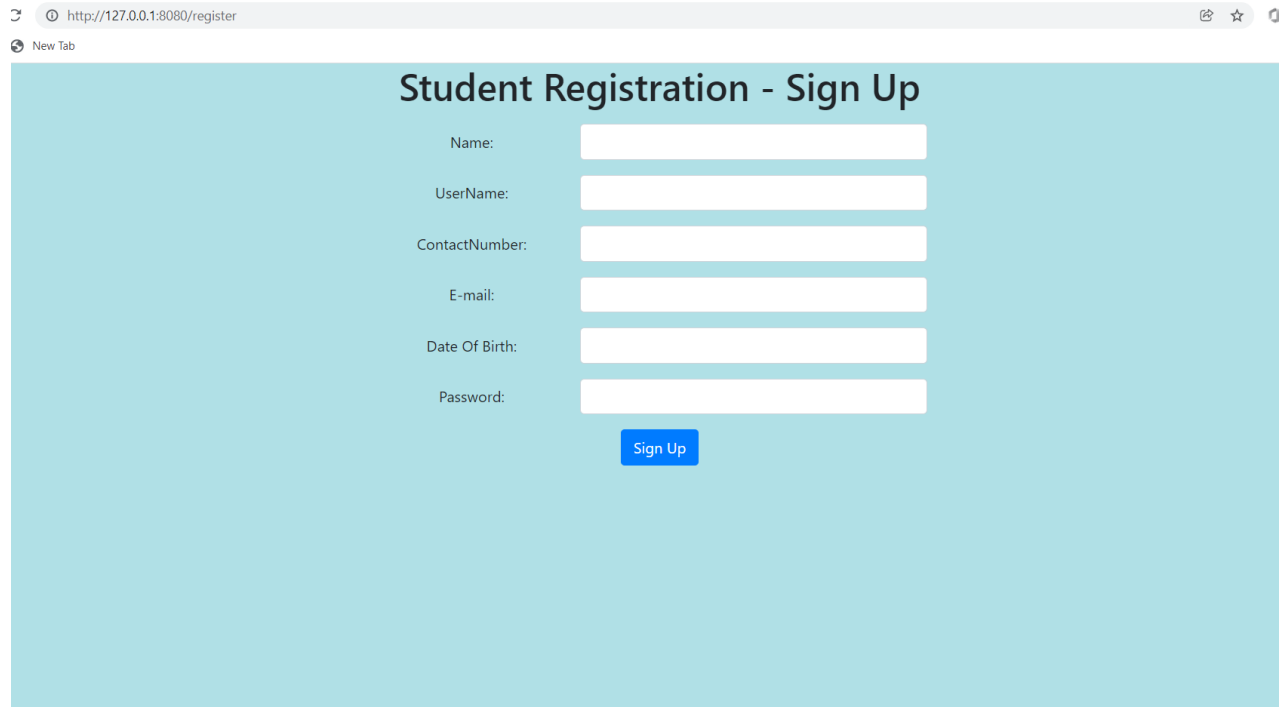
Admin Sign In

UserName:

Password:

[Sign In](#)

Student Registration:



The screenshot shows a web browser window with the address bar displaying "http://127.0.0.1:8080/register". The browser has a "New Tab" tab open. The main content area has a light blue background and features the title "Student Registration - Sign Up" in bold black text. Below the title is a registration form with the following fields and labels:

- Name:
- UserName:
- ContactNumber:
- E-mail:
- Date Of Birth:
- Password:

Below the password field is a blue button with the text "Sign Up".

Registering with data:

← → ↻ http://127.0.0.1:8080/register

Apps New Tab Reading list

Student Registration - Sign Up

Name:

UserName:

ContactNumber:

E-mail:

Date Of Birth:

Password:

[Sign Up](#)

Successful Registration Done.

← → ↻ http://127.0.0.1:8080/process_register

Apps New Tab Reading list

You have signed up successfully!

[Click here for Home Page](#)

Data Added in Database Successfully:

The screenshot displays the SQL Developer environment. On the left, the 'SCHEMAS' pane shows a tree view with 'corejavadb' selected, containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. Below this, the 'Administration' and 'Information' tabs are visible, with 'No object selected' displayed.

The main editor shows a SQL script with the following content:

```

1 • create database corejavadb;
2 • use corejavadb;
3 • create table users(
4   id INT NOT NULL AUTO_INCREMENT,contact_number VARCHAR(100) NOT NULL,name VARCHAR(100) NOT NULL,email VARCHAR(100) NOT NULL,password VARCHAR(100) NOT NULL,
5
6 • use corejavadb;
7 • create table Drive(id INT NOT NULL AUTO_INCREMENT,company_name VARCHAR(100) NOT NULL,drive_date VARCHAR(100) NOT NULL,location VARCHAR(100) NOT NULL,
8   );
9
10 • select *from drive;
11 • select *from users;
12 • use corejavadb;
13 • insert into users values(1,'7619186199','Sahana','ksahana1006@gmail.com','123456','1','15-06-1998','Sahana');
14
15
16

```

Below the script, the 'Result Grid' is displayed, showing the execution results of the 'select *from users;' query. The grid has 8 columns: id, contact_number, name, email, password, is_admin, Date_Of_Birth, and user_name. The data is as follows:

id	contact_number	name	email	password	is_admin	Date_Of_Birth	user_name
1	7619186199	Sahana	ksahana1006@gmail.com	123456	1	15-06-1998	Sahana
2	7892316031	Sahana K	ksahana1006@gmail.com	Gayathri@r	0	15/06/1998	sahana
3	12345678	priyanka	priyanka@gmail.com	123456	0	1-1-2020	priyanka
4	7892316222	chandana	chandana@gmail.com	123456	0	2/2/2002	chandu
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, the 'users 1' tab is active, and the 'Apply' and 'Revert' buttons are visible.

Login With student registered Details:

Student Sign In

UserName: ksahana1006@gmail.com

Password: *****

Sign In

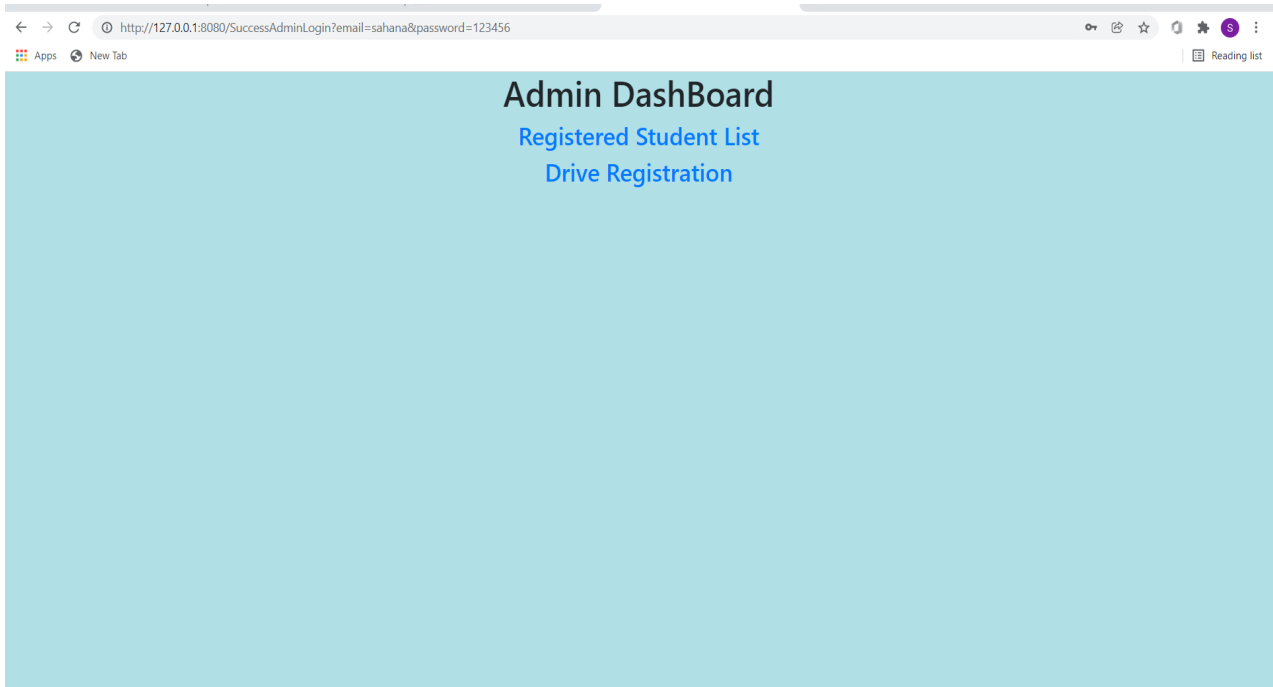
Login with Admin UserID:

Sign Out

List of Drives

Company Name	Location	Date Of Drive
cognizant	BANGALORE	23-01-2022
infosys	chennai	08-12-2021

Admin Dashboard:



Registered student lists:

← → ↻ http://127.0.0.1:8080/users

Apps New Tab

Sign Out

List of Students

E-mail	Name	User name	Contact Number	Date Of Birth
ksahana1006@gmail.com	Sahana K	sahana	7892316031	15/06/1998
priyanka@gmail.com	priyanka	priyanka	12345678	1-1-2020
chandana@gmail.com	chandana	chandu	7892316222	2/2/2002

Reading list

Add drive details:

Taliban alla kannada appu movie x WhatsApp x PDF to Word Converter - 100% F x Placement Management System x +

← → ↻ http://127.0.0.1:8080/drive/edit

Apps New Tab

Drive Registration - Add-Edit

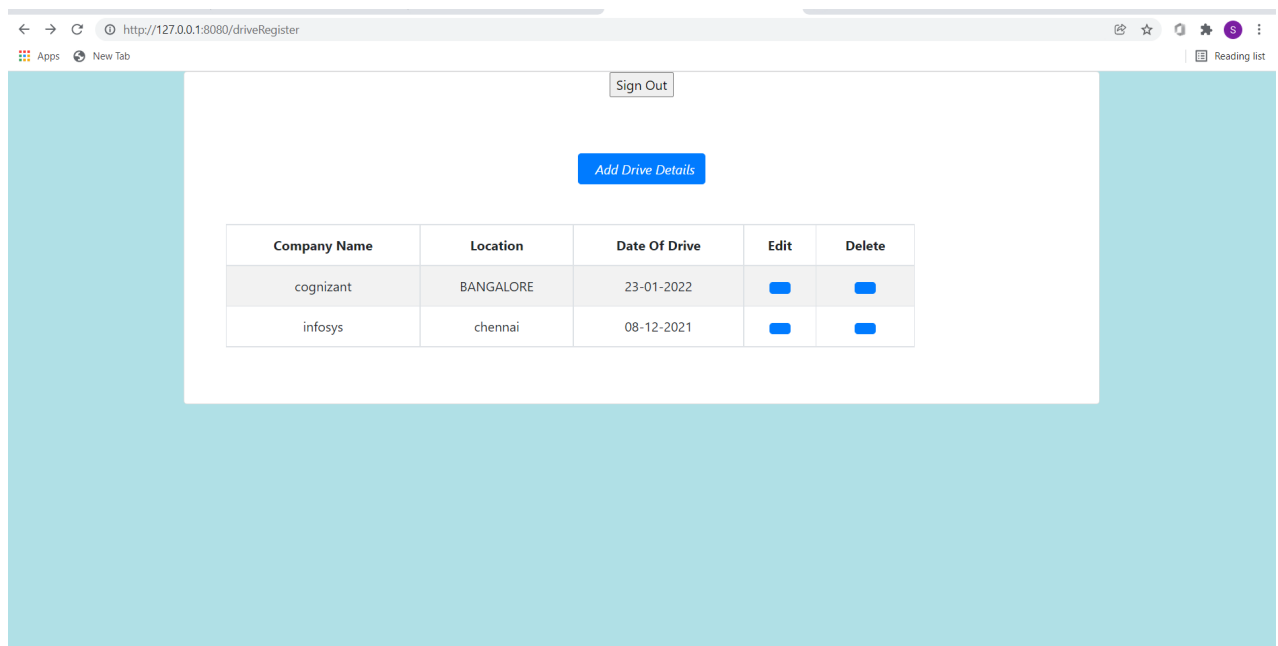
Company Name:

Drive Date:

Location:

Reading list

Successful Added Drives



Thank you