

# EduBridge



A Project

Report On

## **Placement Management System**

By

**Srujana**

**BTECH**

**Batch: 2021 – 5883**

**Center: Bangalore**

Under the Guidance of,

**Miss Mamta Boga.**

**Technical Trainer**

**EduBridge**

(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 three **Modules:**

1. Admin Login
2. Student Registration
3. 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 registerd 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 registration. In Cart, admin can access, they can add delete and modify the jobs if he wants.

## **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

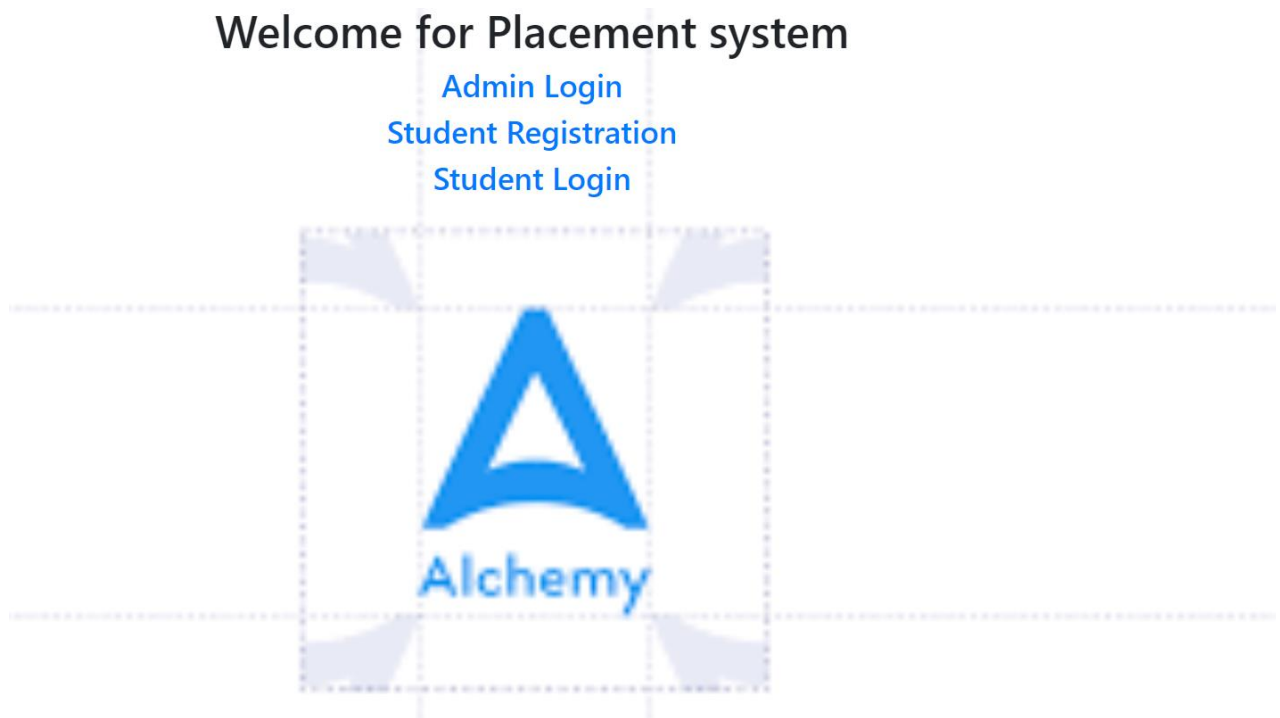
**Operating System:** Window 10.

## **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:



### Admin Login:

The screenshot shows the "Admin Sign In" form. The title "Admin Sign In" is centered at the top in a bold, black font. Below the title, there are two input fields: "UserName:" and "Password:". The "UserName:" field has a blue border and a cursor inside. The "Password:" field has a white border. Below the "Password:" field is a blue button with the text "Sign In" in white. The entire form is set against a light blue background.

## Student Registration:

⌕ http://127.0.0.1:8080/register

New Tab

### Student Registration - Sign Up

Name:

UserName:

ContactNumber:

E-mail:

Date Of Birth:

Password:

## Registering with data:

⌕ http://127.0.0.1:8080/register

Apps New Tab Reading list

### Student Registration - Sign Up

Name:

UserName:

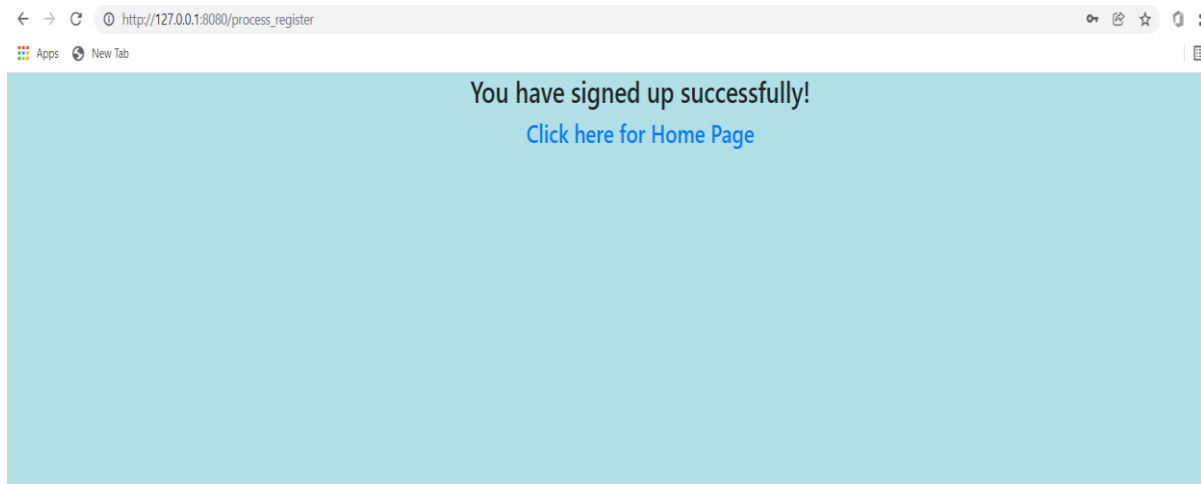
ContactNumber:

E-mail:

Date Of Birth:

Password:

## Successful Registration Done.



## Data Added in Database Successfully:

A screenshot of a database management tool interface. The left sidebar shows a tree view of schemas, including `bookauthor`, `corejavadb` (selected), `edu`, and `sys`. The `corejavadb` schema is expanded, showing `Tables`, `Views`, `Stored Procedures`, and `Functions`. The main area displays a list of SQL queries in a text editor. The queries are as follows:

```
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,is_admin INT NOT NULL,Date_Of_Birth DATE NOT NULL,user_name 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');
```

Below the queries, a "Result Grid" is displayed, showing the results of the last query. The grid has columns for `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
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The bottom of the interface shows a "users 1 x" tab and buttons for "Apply" and "Revert".

## Login With student registered Details:

Student Sign In

UserName: ksahana1006@gmail.com

Password: \*\*\*\*\*

Sign In

## Login with Admin UserID:

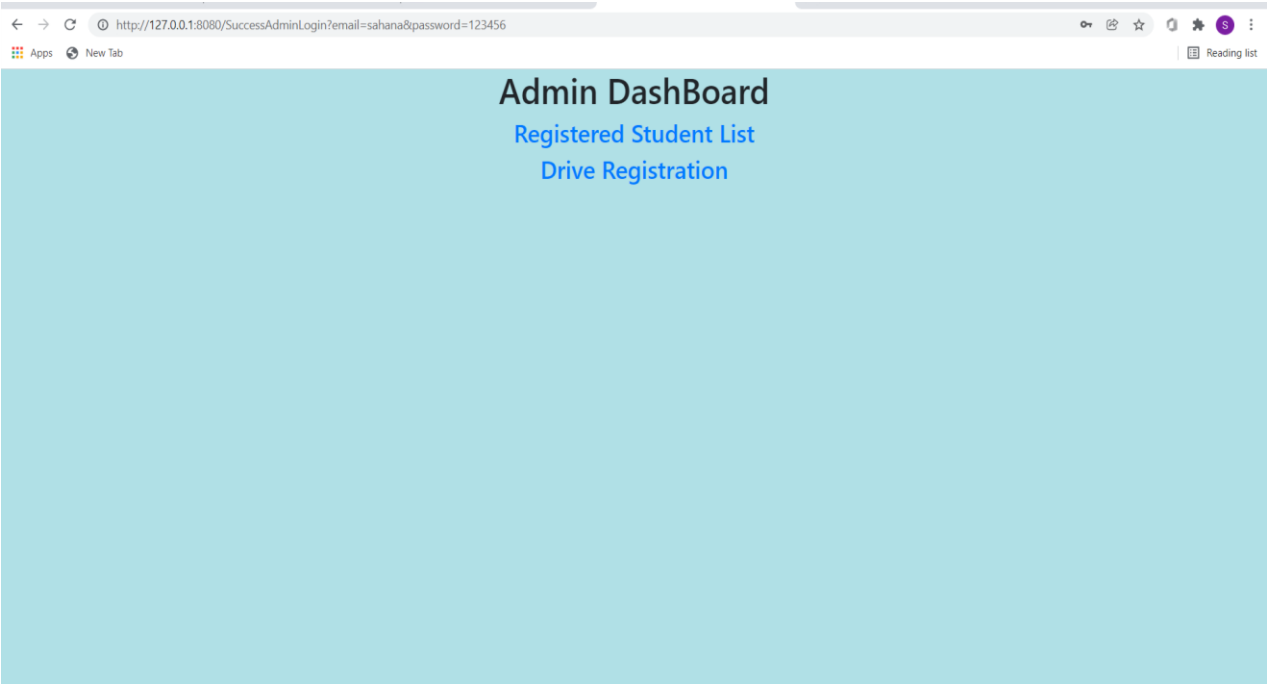
Sign Out

List of Drives

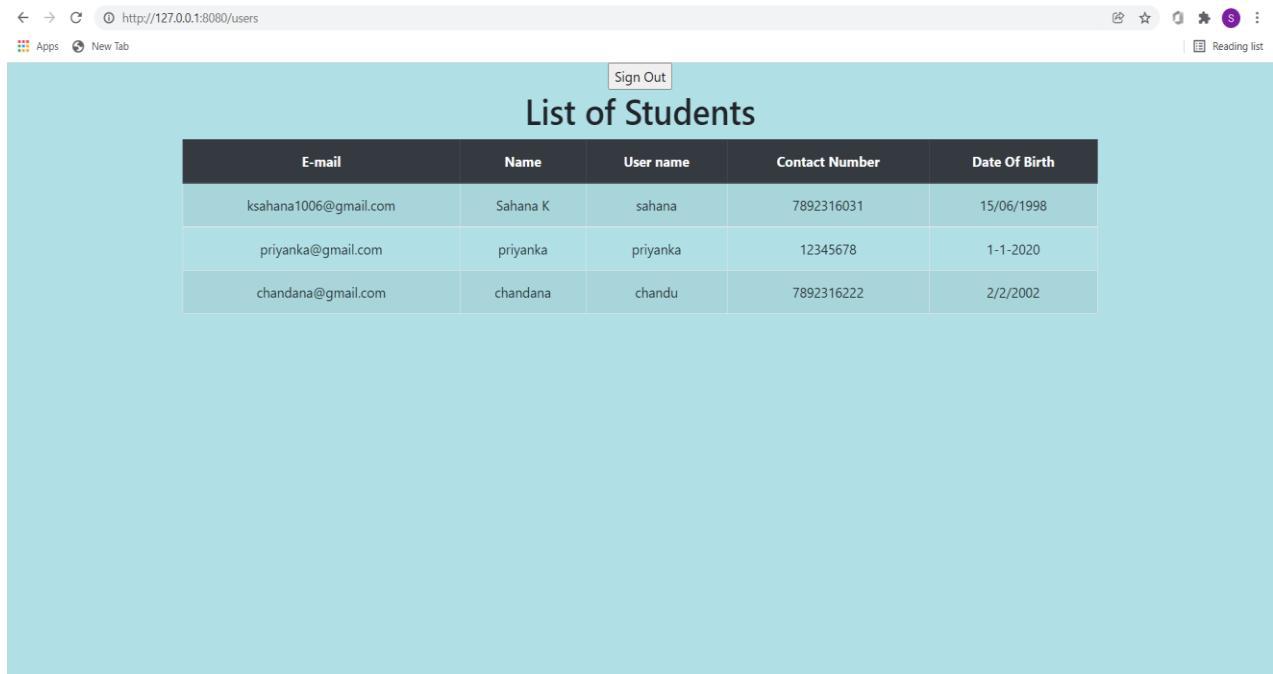
Company Name	Location	Date Of Drive
cognizant	BANGALORE	23-01-2022
infosys	chennai	08-12-2021



# Admin Dashboard:



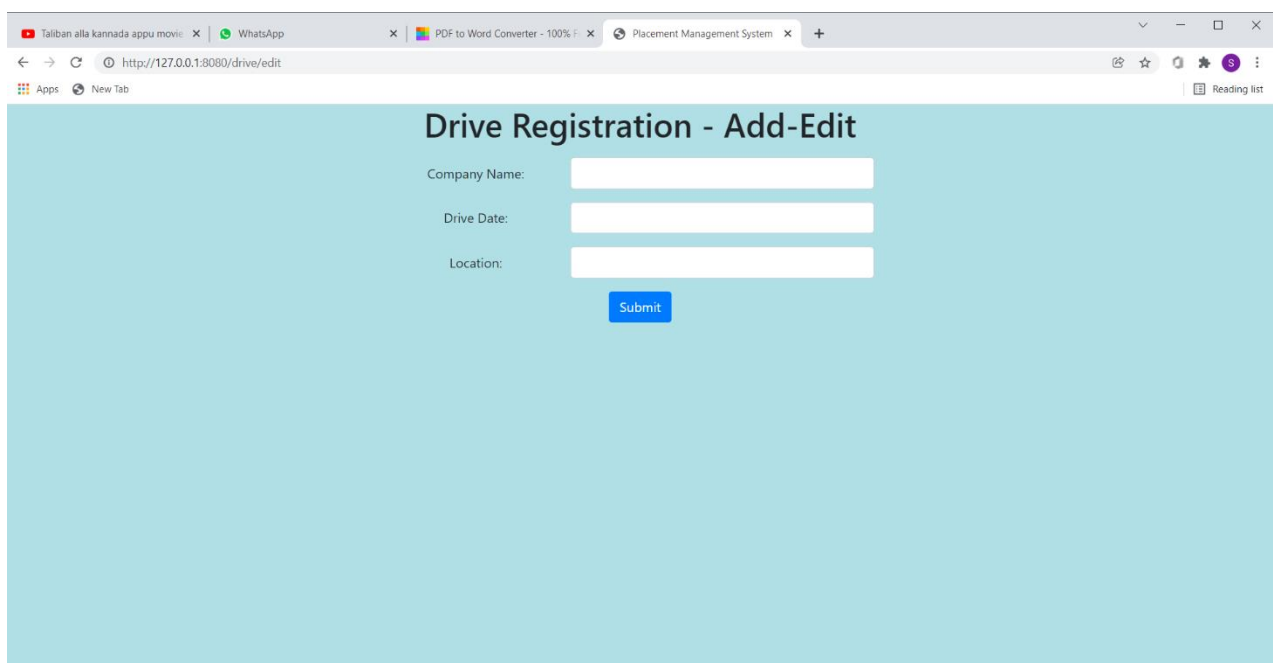
## Registered student lists:



The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8080/users`. The page has a light blue background. At the top right, there is a 'Sign Out' button. The main heading is 'List of Students'. Below it is a table with five columns: 'E-mail', 'Name', 'User name', 'Contact Number', and 'Date Of Birth'. The table contains three rows of student data.

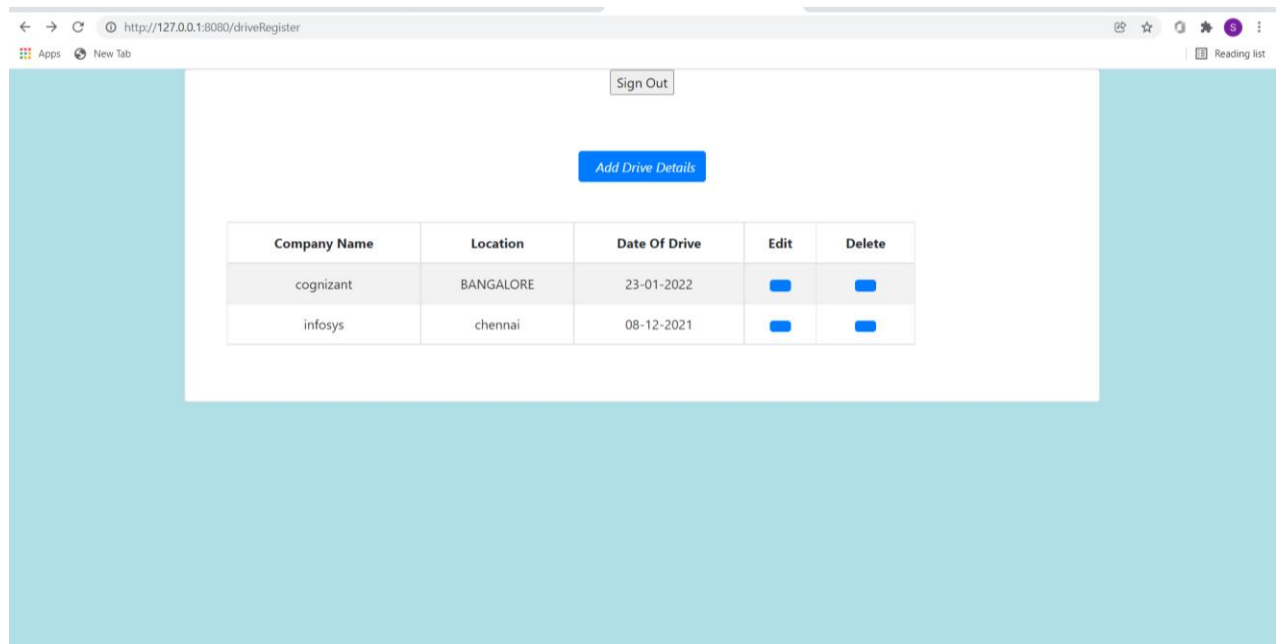
E-mail	Name	User name	Contact Number	Date Of Birth
ksahana1006@gmail.com	Sahana K	sahana	7892316031	15/06/1998
priyanka@gmail.com	priyanka	priyanka	12345678	1-1-2020
chandana@gmail.com	chandana	chandu	7892316222	2/2/2002

## Add drive details:



The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8080/drive/edit`. The page has a light blue background. The main heading is 'Drive Registration - Add-Edit'. Below the heading is a form with three input fields: 'Company Name:', 'Drive Date:', and 'Location:'. Each field is represented by a white rectangular box. Below the 'Location' field is a blue 'Submit' button.

## Successful Added Drives



# Thank you