

# **CE377 - ADVANCED WEB TECHNOLOGY**

## **ASSIGNMENT**

Student ID: 19CE088

Student Name: Vraj Parikh

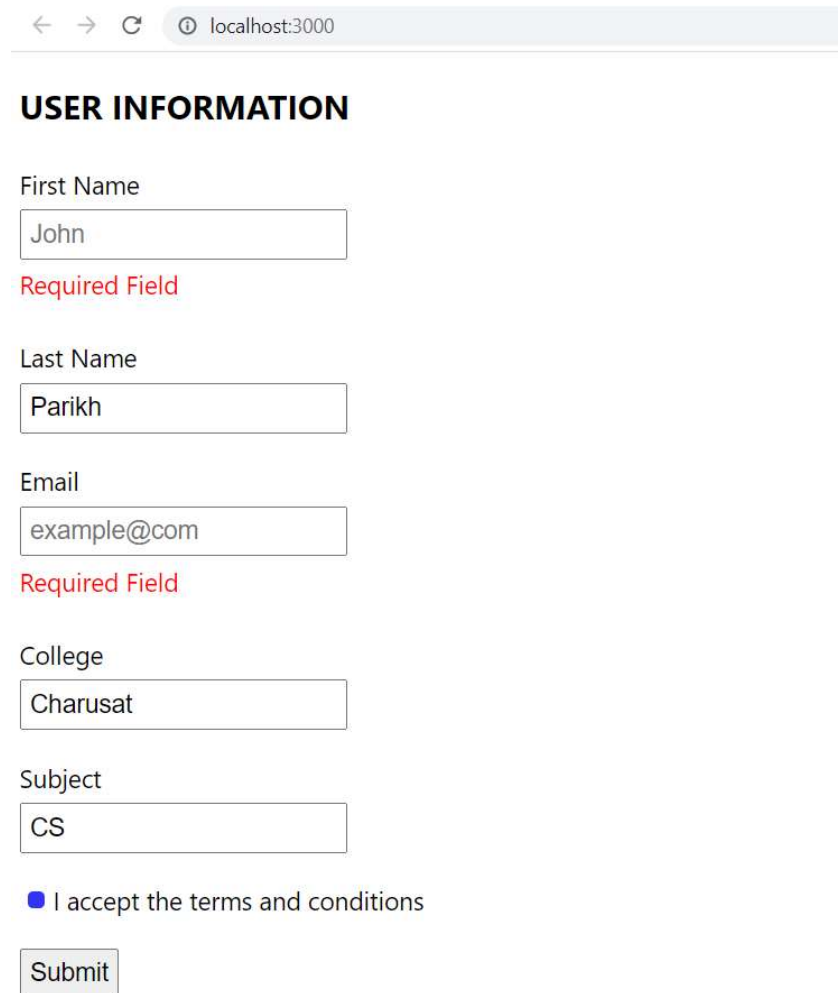
## ASSIGNMENT - 1

**AIM:** React Form Handling

**GITHUB LINK:**

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%20%20-%20React%20Form%20Handling>

**IMPLEMENTATION:**



← → ↻ ⓘ localhost:3000

### USER INFORMATION

First Name  
  
Required Field

Last Name

Email  
  
Required Field

College

Subject

☒ I accept the terms and conditions

## ASSIGNMENT - 2

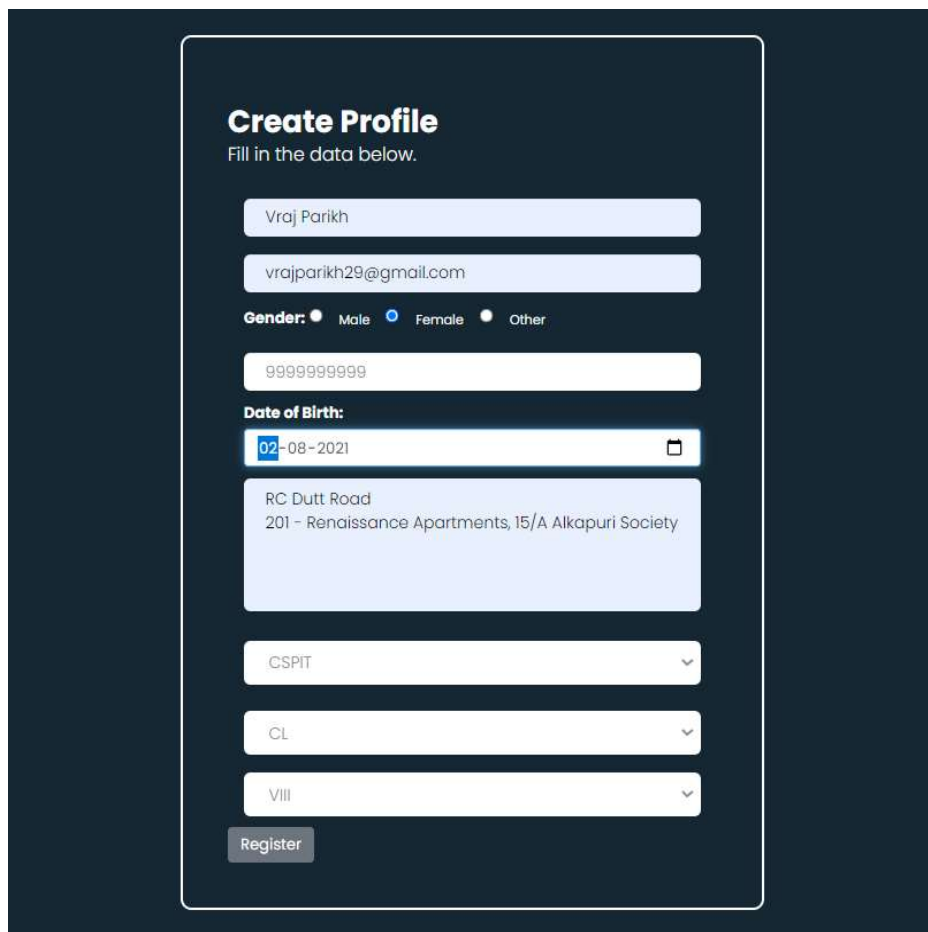
**AIM:** Create Profile page with proper validation and form-control for following information using NodeJs and Bootstrap only.

1. First Name and Last name
2. Email-id
3. Mobile Number
4. Date of Birth
5. Gender
6. Address
7. Select Institute
8. Select Department
9. Select Semester
10. Display all information in HTML Table.

### GITHUB LINK:

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%20%20-%20NodeJS%20Bootstrap%20Form>

### IMPLEMENTATION:



The image shows a 'Create Profile' form on a dark blue background. The form is titled 'Create Profile' and includes the instruction 'Fill in the data below.' The form fields are as follows:

- First Name and Last name:** A text input field containing 'Vraj Parikh'.
- Email-id:** A text input field containing 'vrajparikh29@gmail.com'.
- Gender:** Radio buttons for 'Male', 'Female' (selected), and 'Other'.
- Mobile Number:** A text input field containing '9999999999'.
- Date of Birth:** A date picker showing '02-08-2021'.
- Address:** A text area containing 'RC Dutt Road' and '201 - Renaissance Apartments, 15/A Alkapuri Society'.
- Select Institute:** A dropdown menu showing 'CSPIT'.
- Select Department:** A dropdown menu showing 'CL'.
- Select Semester:** A dropdown menu showing 'VIII'.
- Register:** A button at the bottom left of the form.

## Created Profile

|               |   |
|---------------|---|
| Full Name     | Vraj Parikh   |
| Email         | vrajparikh29@gmail.com  |
| Mobile        | 9999999999  |
| Date of Birth | 2021-08-02  |
| Gender        | Male  |
| Address       | RC Dutt Road<br>201 - Renaissance Apartments, 15/A Alkapuri Society |
| Institute     | CSPIT   |
| Department    | CE  |
| Semester      | V   |

## ASSIGNMENT - 3

**AIM:** Create SignIn and Signup Page layout using Handlebars.

**GITHUB LINK:**

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%20%20-%20Handlerbars>

**IMPLEMENTATION:**

### Signin

The screenshot shows a web browser window at localhost:3000/signup. The page has a green background with a central dark green login form. The form contains a message "New User registered." in green, followed by input fields for "Email" (containing "vraj") and "Password". Below these is a green "LOGIN" button. At the bottom of the form, it says "Not yet registered? Sign Up Now!". The page header says "This Is The Header Login" and the footer says "This Is The Footer".

### Signup

The screenshot shows a web browser window at localhost:3000/signup. The page has a green background with a central dark green signup form. The form contains a message "User Already Exists" in red, followed by input fields for "Email" (containing "sugarbact51@gmail.com"), "Email" (containing "vraguarkh@gmail.com"), "Password", and "Confirm Password". Below these is a green "SIGN UP" button. At the bottom of the form, it says "Already have an Account? Login Now!". The page header says "This Is The Header Sign Up" and the footer says "This Is The Footer".

Home



## **ASSIGNMENT - 4**

**AIM:** To retrieve records from MySQL and display it on console.

**GITHUB LINK:**

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%20%20-%20NodeJS%20MySQL>

**IMPLEMENTATION:**

```
Server is running on port number 3000
Connected successfully to MySql server

List of Students :

102
Mrugendra Rahevar
123456789

103
Martin Parmar
123456789

104
Vraj Shah
123456789
```

## ASSIGNMENT - 5

### AIM:

Task1: Using Handlebar, create HTML page that contains following information

1. Subject-Code, Subject-Name (Text-box)
2. Institute-Name (Text-box)
3. Department-Name (Text-box)
4. Semester=>Dropdown menu

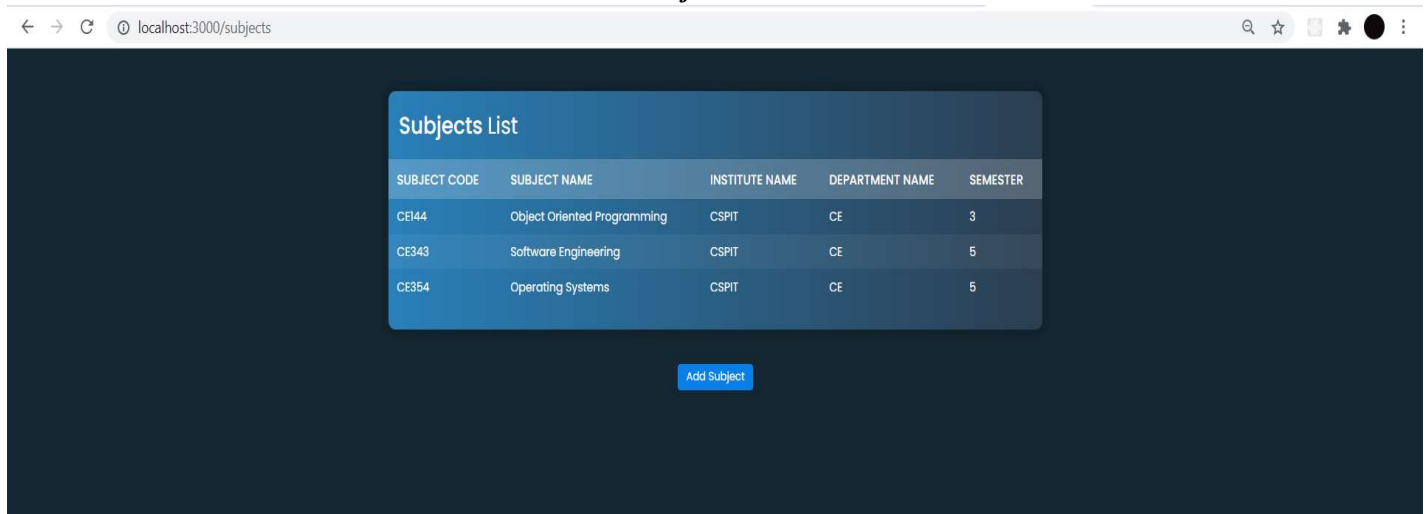
Task2: Store above information into MySQL database and display

### GITHUB LINK:

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%205%20-%20NodeJS%20DB%20Operation>

### IMPLEMENTATION:

#### *Subjects List*



| SUBJECT CODE | SUBJECT NAME                | INSTITUTE NAME | DEPARTMENT NAME | SEMESTER |
|--------------|-----------------------------|----------------|-----------------|----------|
| CE144        | Object Oriented Programming | CSPIT          | CE              | 3        |
| CE343        | Software Engineering        | CSPIT          | CE              | 5        |
| CE354        | Operating Systems           | CSPIT          | CE              | 5        |

Add Subject



***Subject Form***

A screenshot of a web browser window displaying a form titled "Add Subject". The browser's address bar shows "localhost:3000/subject". The form is centered on a dark blue background and is enclosed in a white border. It contains the following elements:

- Add Subject**: The main title of the form.
- Fill in the data below.**: A subtitle instruction.
- Subject Code**: A text input field.
- Subject Name**: A text input field.
- Institute Name**: A text input field.
- Department Name**: A text input field.
- SEMESTER**: A dropdown menu with a downward arrow.
- Register**: A blue button.
- View Subjects**: A blue button.

## ASSIGNMENT - 6

### AIM:

Create html form using bootstrap with following information

1. Course-id, Course Name [text box]
2. Department Name [Radio Button]
3. Institute Name [Check Box]
4. University Name [Drop Down]

Display all Record into dynamic table with edit and delete operation.

### GITHUB LINK:

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%206%20-%20CRUD%20with%20Express%2C%20Bootstrap%20and%20MySQL>

### IMPLEMENTATION:

#### *Home Page*

← → ↻ ⓘ 127.0.0.1:5501/client/index.html ☆ 📄 🌐 ⚙️ 🔍

## CRUD Operation Using ExpressJS and Mysql

Course Id:

Course Name:

Department: ☐ CE ☐ IT ☐ CL ☐ ME ☐ CSE ☐ EC

Institute: ☐ CSPIT ☐ DEPSTAR

University:

| ID      | Name | Department | Institute | University | Delete | Edit |
|---------|------|------------|-----------|------------|--------|------|
| No Data |      |            |           |            |        |      |

View Courses

127.0.0.1:5501/client/index.html

CRUD Operation Using  
ExpressJS and Mysql

Course Id:

Course Name:

Department: ☐ CE ☐ IT ☐ CL ☒ ME ☐ CSE ☐ EC

Institute: ☒ CSPIT ☒ DEPSTAR

University:

Add Course

search by name

Search by User Name

| ID    | Name            | Department | Institute      | University          | Delete | Edit |
|-------|-----------------|------------|----------------|---------------------|--------|------|
| CE355 | Data Structures | CE         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |
| CE266 | Programming     | IT         | CSPIT          | Others              | Delete | Edit |
| ME234 | Machines        | ME         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |

Filter Courses

127.0.0.1:5501/client/index.html

# CRUD Operation Using ExpressJS and Mysql

Course Id:

Course Name:

Department: ☐ CE ☐ IT ☐ CL ☒ ME ☐ CSE ☐ EC

Institute: ☒ CSPIT ☒ DEPSTAR

University:

Add Course

Data

Search by User Name

| ID    | Name            | Department | Institute      | University          | Delete | Edit |
|-------|-----------------|------------|----------------|---------------------|--------|------|
| CE355 | Data Structures | CE         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |

**Delete Courses**

← → ↻ ⓘ 127.0.0.1:5501/client/index.html ☆ 🏠 ⚙️ 🔍

## CRUD Operation Using ExpressJS and Mysql

Course Id:

Course Name:

Department: ☐ CE ☐ IT ☐ CL ☐ ME ☐ CSE ☐ EC

Institute: ☐ CSPIT ☐ DEPSTAR

University:

Add Course

search by name

Search by User Name

| ID    | Name            | Department | Institute      | University          | Delete | Edit |
|-------|-----------------|------------|----------------|---------------------|--------|------|
| CE355 | Data Structures | CE         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |
| ME234 | Machines        | ME         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |

Update Courses

127.0.0.1:5501/client/index.html

Course Id:

Course Name:

Department: ☐ CE ☐ IT ☐ CL ☐ ME ☐ CSE ☐ EC

Institute: ☐ CSPIT ☐ DEPSTAR

University: 

Please select a university

Add Course

search by name

Search by User Name

| ID    | Name            | Department | Institute      | University          | Delete | Edit |
|-------|-----------------|------------|----------------|---------------------|--------|------|
| CE355 | Data Structures | CE         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |
| ME234 | Machines        | ME         | CSPIT, DEPSTAR | Charotar University | Delete | Edit |

Course Id:

Course Name:

Department: ☐ CE ☐ IT ☐ CL ☐ ME ☐ CSE ☐ EC

Institute: ☐ CSPIT ☐ DEPSTAR

University: 

Please select a university

Update

## ASSIGNMENT - 7

### AIM:

Create Database REST API for login application

1. post('/signup') route create user in to database(Mysql or MongoDB)
2. Use salt and hashing concept to secure password.
3. get('/login/') route verify credential with database and display success or failure message in the console.

### GITHUB LINK:

<https://github.com/vraj291/CE377-AWT-Assignments/tree/main/Assignment%20%20-%20Hashing>

### IMPLEMENTATION:

*/signup*

The screenshot displays a REST client interface with the following details:

- URL:** http://localhost:4000/signup
- Method:** POST
- Body (JSON):**

```
1 {
2   "name": "Vraj Parikh",
3   "email": "test1@gmail.com",
4   "password": "Hello123"
5 }
```
- Status:** 200 OK
- Time:** 1346 ms
- Size:** 280 B
- Response Body (JSON):**

```
1 {
2   "err": false,
3   "msg": "Successfully signed up."
4 }
```

*/login*

http://localhost:4000/login

Save

GET http://localhost:4000/login Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "email": "test1@gmail.com",
3   "password": "Hello123"
4 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 395 ms Size: 459 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "err": false,
3   "data": {
4     "_id": "6165162329bb8d869803e5bf",
5     "name": "Vraj Parikh",
6     "email": "test1@gmail.com",
7     "hashed_password": "$2b$12$Rewb8CBsve@YKIIBYqk/90ne154ho7g0t0Czk6rUd3yZ4S3IgPbNu",
8     "__v": 0
9   },
10   "msg": "Successfully logged in."
11 }
```