

## **INDEX**

CONTENT	PAGE NO
Module and Concept used	3
Description of Modules and package used	4-5
Step by step Interaction with user	6-10
Step 1: Landing page	6
Step 2: SignUp	7
Step 3: SignIn	8
Step 4: LogOut	9

## **CONCEPT AND MODULE USED**

---

*Node Js*

*Express Js*

*Bcrypt js*

*Express session*

*Cookies*

*Path*

*Connect-mongodb-sessions*

*Mongoose*

*Mongodb*

---

## DESCRIPTION

### **Bcrypt Js:**

*We used this module for hashing our passwords so that our password is protected and secure in the Database while its stored.*

We used two major methods of this which are “.compare”, “.hash”

```
const hashedPsw = await bcrypt.hash(password, 12);
```

```
const isMatch = await bcrypt.compare(password, user.password);
```

### **Sessions:**

We made a session for the valid user and destroyed it while he logged out.

```
app.use(
  session({
    secret: "For the Webtech DA",
    resave: false,
    saveUninitialized: false,
    store: store,
  })
);
```

```
req.session.destroy((err)=>
{
  if(err) throw err;
  res.redirect('/')
})
```

```
req.session.isAuthenticated = true;
```

### **Cookies :**

We created Cookies, with the help of node js and Express js.

One cookie that was created is Session Id for the session which we created , when the user logged in Successfully.

### **Connect-mongodb-session:**

We used this module for the linking of *Mongodb* with our sessions. With this, a link was established between Database and the Session created for the *authentication* purposes.

### **Mongoose:**

We used the mongoose to interact with our database Dynamically with the use of our node js and express js file and logic.

We created a user schema for the Signup Page that will extract the content specified in that.

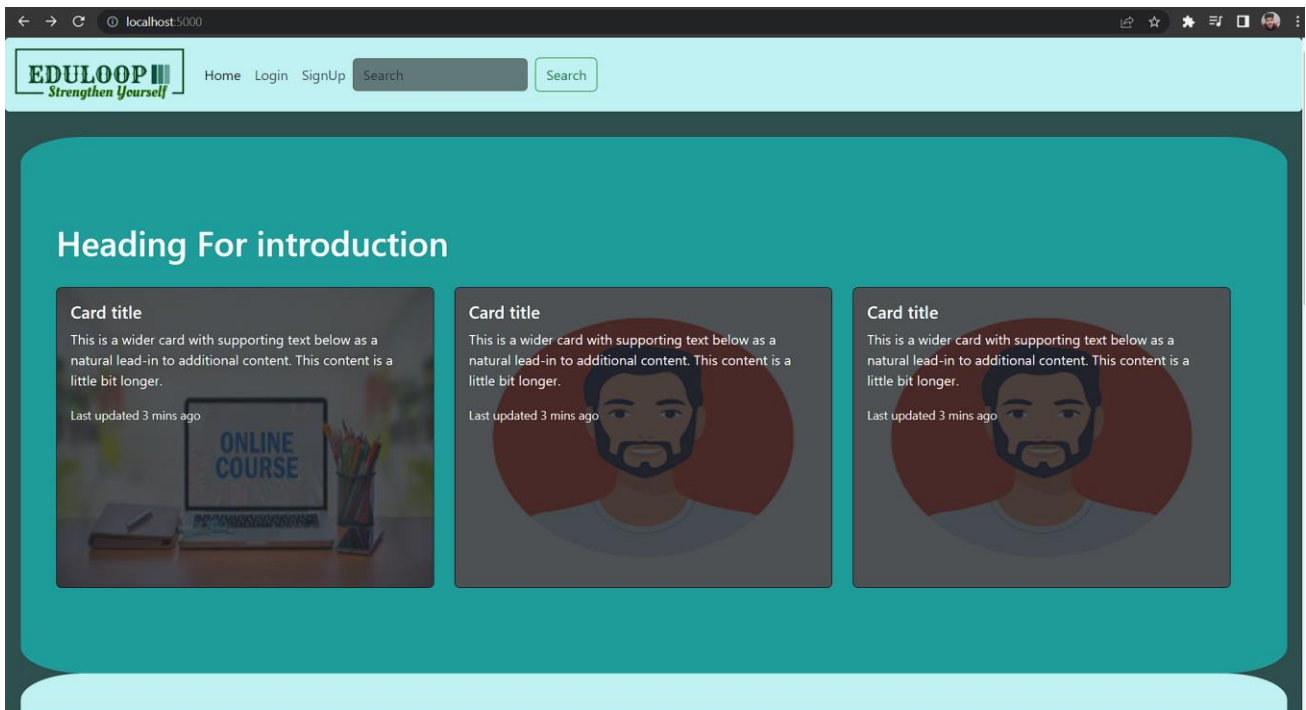
```
const userSchema = new schema({.....})
```

## HOW WILL USER INTERACT?:

### Step 1:

User will be landed on the Main page, then we will click for Signup(new user) or Login( if already user):

Suppose he Clicked on SignUp now he will be directed to the signup page:



### Step 2:

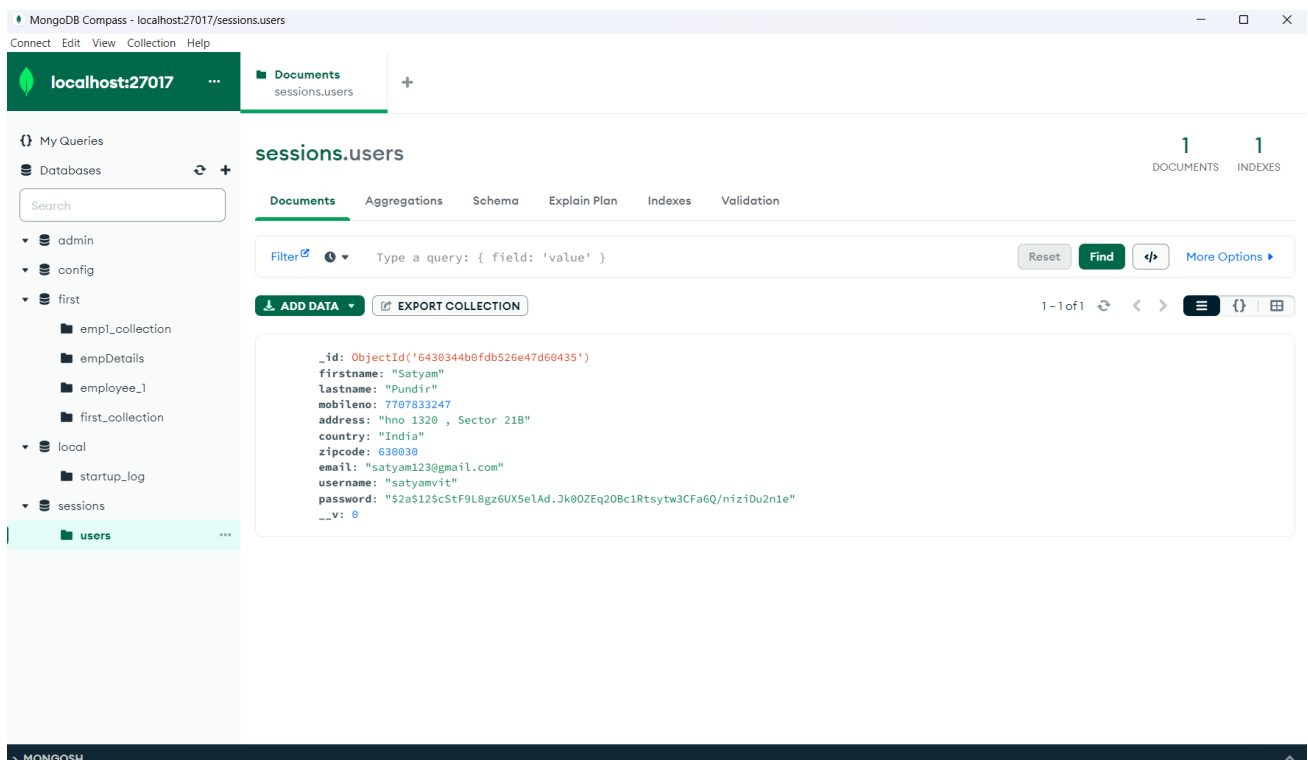
On Signup we will enter the required details and the details will be stored automatically in the database.

A screenshot of the Eduloop signup page. The browser address bar shows 'localhost:5000/signup'. The page has a teal background. On the left, there is a graphic with various social media icons (Instagram, Twitter, Facebook, YouTube, Spotify, TikTok) arranged around a central logo. On the right, there is a white form with the following fields: 'First Name' (filled with 'Satyam'), 'Last Name' (filled with 'Pundir'), 'Mobile No' (filled with '7707833247'), 'Address' (filled with 'hno 1320, Sector 21B'), 'Country' (filled with 'India'), 'City' (filled with 'Chandigarh'), 'Punjab' (selected in a dropdown), 'Zip' (filled with '630030'), 'Email address' (filled with 'satyam123@gmail.com'), 'Username' (filled with 'satyamvit'), 'Password' (masked with dots), and 'Re-enter Password' (masked with dots). Below the form, there is a checkbox for 'Send me updates via Email\*' which is unchecked. At the bottom, there are two buttons: 'Sign Up' for new users and 'Sign In' for existing users. The text '\* New User' is above the 'Sign Up' button, and '\* Already a User' is above the 'Sign In' button.

New Sessions Database is created with a collection of users, containing our details.


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE


[nodemon] starting `node server.js`
MongoDB connected
entered the post of sigup
Satyam
[]
```



### Step 3:

Now we will be redirected to the Login Page, now If we enter the correct value for the login we will be logged in and a user session will be created with a cookie in it.





### Login


UserName


Email address

Password

[\\* New User](#)

[\\* Already a User](#)





### Login

UserName

Email address

Password

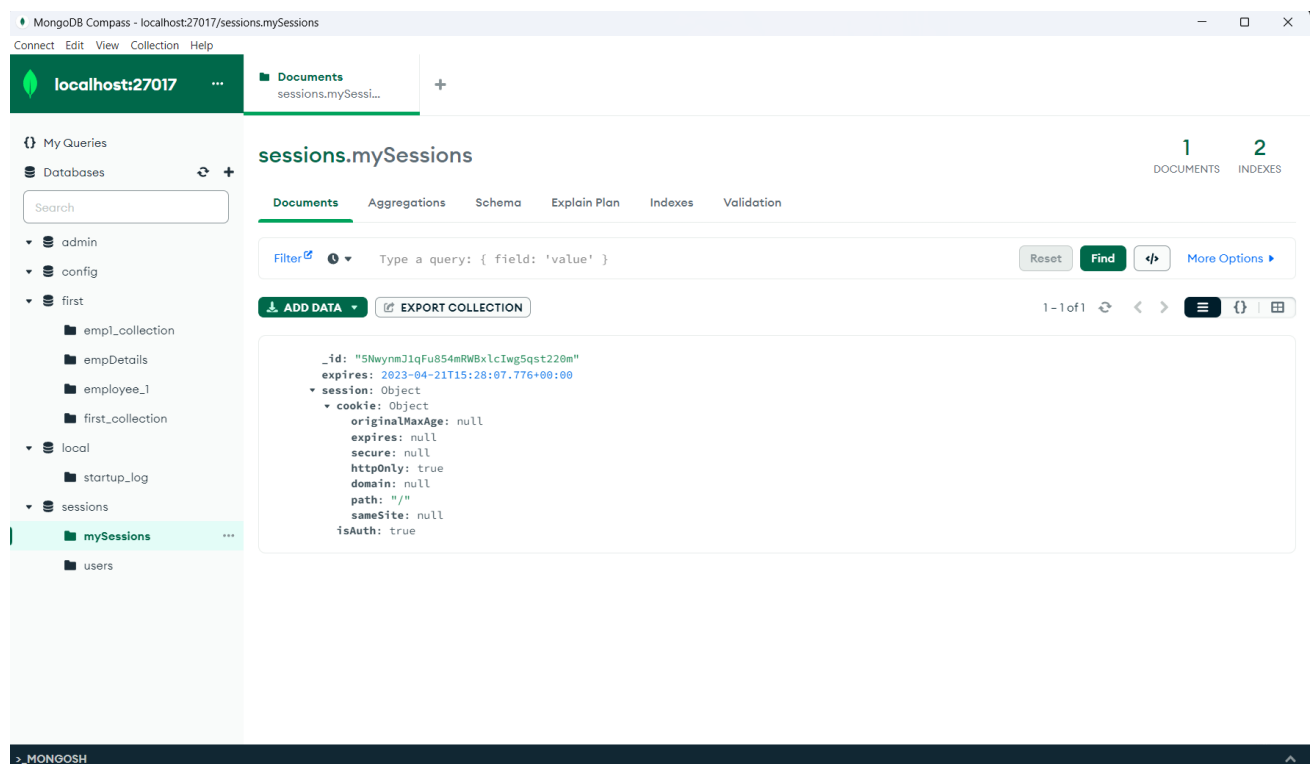
[\\* New User](#)

[\\* Already a User](#)


## Step 4:

When the user has entered the correct password then they will be directed to our main internal content where they can access material.

A session is created with the `isAuth` variable set to `true`, that means this session is validated for the user and he can now only access the main internal content.






[Home](#)
[Login](#)
[SignUp](#)
[LOGOUT](#)

Engineering

Physics

Maths

C++

JAVA

DSA

## Engineering

Engineering is the field of study and application of scientific, mathematical, and technical principles to design, develop, test, and improve products, systems, and processes. Engineers use their knowledge to create solutions to practical problems and to improve existing technologies. Engineering is a broad field that encompasses various disciplines and includes,

- Mechanical Engineering
- Electrical Engineering
- Civil Engineering
- Aerospace Engineering
- Chemical Engineering
- Computer Engineering
- and many more.

Each discipline focuses on specific areas of expertise and uses different tools, methods, and techniques to solve problems and develop new products. Engineers work in a variety of settings, including research and development laboratories, manufacturing facilities, government agencies, and consulting firms. They often collaborate with other professionals, such as scientists, technicians, and managers, to bring their ideas to fruition and ensure the success of their projects.

### Advantages of opting for engineering in current times:

- I. High demand: Engineering is a field that is always in high demand due to its diverse applications and contributions to various industries such as technology, manufacturing, construction, transportation, and healthcare.
- II. Career prospects: Engineering provides a wide range of career opportunities with a high earning potential. Graduates can work in various fields, including research and development, design, construction, management, and consulting.
- III. Innovation: Engineers are at the forefront of innovation, developing new products, processes, and technologies that can revolutionize

Now user has the access to the content until they manually logout. He can navigate to the login via clicking to login, or he can go to the home by clicking on the home or the LOGO.

When he clicks on Logout it will be Logged out and session will be destroyed.

If he tries to access the Main Content Page, they will be presented with Login page since the *session* has been *terminated*. And they need to *re-authorise* the session from Login.

### Code For User Schema is in *user.js* File:

```
const mongoose = require("mongoose");
const schema = mongoose.Schema;

const userSchema = new schema({
  firstname:
  {
    type: String,
    required:true,

  },
  lastname:
  {
    type: String,
    required:true,

  },
  mobileno:
  {
    type: Number,
    required:true,

  },
  address:
  {
    type: String,

  },
  country:
  {
    type: String,
    required:true,

  },
  country:
  {
    type: String,
    required:true,

  },
  zipcode:
  {
    type: Number,
    required:true,

  },
},
```

```

    email:
    {
        type: String,
        required: true,
        unique: true,
    },
    username: {
        type: String,
        required: true,
        unique: true,
    },
    password:{
        type: String,
        required: true,
    },
    repassword:{
        type: String,

    }
});

module.exports = mongoose.model('User',userSchema);

```

### **Code for all the backend is stored in Server.js**

```

const express = require("express");
const bcrypt = require('bcryptjs');
const session = require("express-session");
const path = require('path');
// const fs = require('fs');

const mongoose = require("mongoose");
const MongoDBSession = require('connect-mongodb-session')(session);
const mongouri = "mongodb://127.0.0.1:27017/sessions"
const app = express();

const UserModel = require('./models/user');
const { render } = require("ejs");

app.use(express.static(__dirname + '/Images'));

mongoose
    .connect(mongouri, {

```

```

        useUrlParser: true,
        useUnifiedTopology: true,

    })
    .then((res) => {
        console.log("MonogoDB connected");
    });

const store = new MongoDBSession({
    uri: mongouri,
    collection: "mySessions"
});

// app.set("view engine", "ejs");
app.use(express.urlencoded({ extended: true }));

const isAuth = (req,res,next)=>
{
    if(req.session.isAuth){
        next();
    }

    else{
        res.redirect('/login');
    }
}

app.use(
    session({
        secret: "For the Webtech DA",
        resave: false,
        saveUninitialized: false,
        store: store,
    })
);

// for the main page
app.get("/", (req, res) => {
    res.sendFile(path.join(__dirname + "/views/" + "front_mail.html"))
});

// for the login page
app.get("/login", (req, res) => {
    res.sendFile(__dirname + "/views/" + "login.html")
});

```

```

});

// logic for authentication in login
app.post("/login", async (req, res) => {

    console.log('entered the post method of the login page');
    const { username,email, password } = req.body;

    const user = await UserModel.findOne({ username });
    if (!user) {
        return res.redirect("/login");
    }
    const isMatch = await bcrypt.compare(password, user.password);
    if (!isMatch) {
        return res.redirect("/login");
    }

    // SETS A SESSION FOR IS AUTHENTICATED AS TRUE
    req.session.isAuth = true;
    res.redirect("/eng");

});

// for the sign up process
app.get("/sign_up", (req, res) => {
    res.sendFile(__dirname + "/views/" + "sign_up.html")
});

// for the signup logic
app.post("/sign_up", async (req, res) => {
    const {
        firstname,lastname,mobileno,address,country,city,state,zipcode,email,username,password,repasword
    } = req.body;
    console.log("entered the post of sigup")
    console.log(firstname);
    let user = await UserModel.findOne({ username });

    if (user) {
        return res.redirect('/sign_up');
    }

    const hashedPsw = await bcrypt.hash(password, 12);

    user = new UserModel({
        firstname,
        lastname,
        mobileno,
        address,

```

```

        country,
        city,
        state,
        zipcode,
        email,
        username,
        password: hashedPsw,
    });

    await user.save();
    res.redirect("/login");
});

app.get('/eng', isAuth, (req, res) => {

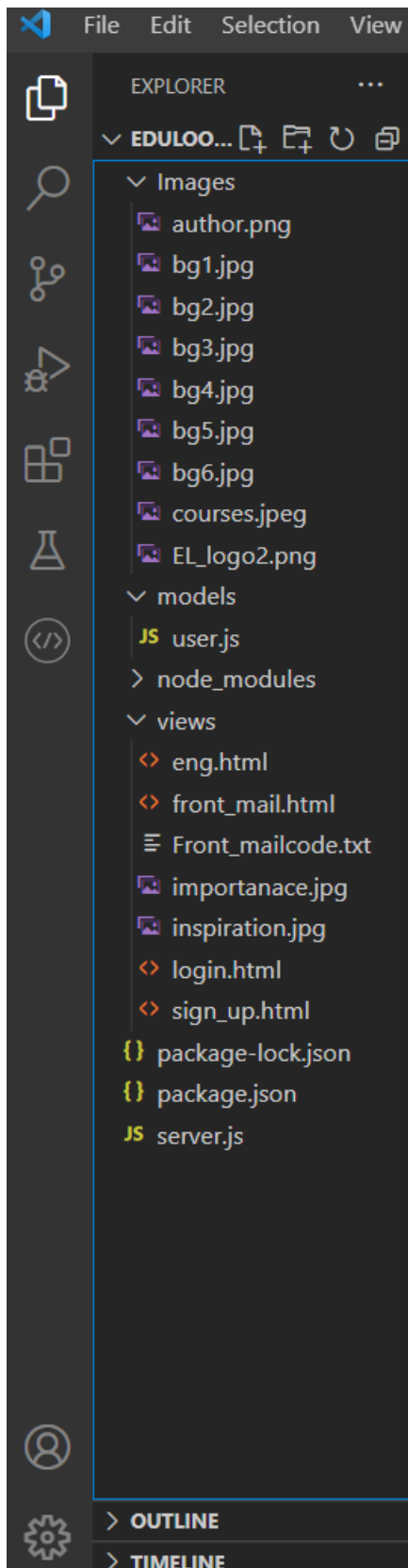
    res.sendFile(__dirname + "/views/" + "eng.html")
});

app.get('/logout', (req, res)=>
{
    req.session.destroy((err)=>
    {
        if(err) throw err;
        res.redirect('/')
    })
})

app.listen(5000);

```

## **File Organisation in VsCode:**



## Code In the VIEWS:

### Code for Login Website:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EduLoop | Login</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
qKXV1j0HvMUeCBQ+QVp7JcfG1760yU08IQ+GpUo5h1bpg51QRiuqHAJz8+BrxE/N"
    crossorigin="anonymous"></script>

  <style>
    nav {

      background-color: #c0f2f3;

      margin-bottom: 30px;

    }

    body {

      background-color: #1e9c99;

    }

    .form-floating:not(.form-control:disabled)::before {
      height: 0;
    }

    .Form_area {
      background-color: rgba(0, 0, 0, 0.2);
      padding: 30px;
      border: 1px solid black;
      border-radius: 10%;
```



```

}

.content {
  margin-top: 10px;
  padding: 50px;
}

label {
  padding: 5px;
}

.image_slider {
  box-shadow: 1cm black;
}

img {
  /* margin-left: 10px; */
  border-radius: 10px;
}
</style>
<script>

var current = 0;

function nextImg() {
  var x = document.getElementById("image");
  var imgs = ["bg1.jpg", "bg2.jpg", "bg3.jpg", "bg4.jpg", "bg5.jpg", "bg6.jpg"];

  if (current < 5) {
    current = current + 1;
  }
  else { current = 0; }

  x.src = imgs[current];
}

setInterval(nextImg, 3500);

</script>
</head>

<body>
  <nav class="navbar navbar-expand-sm">
    <div class="container-fluid">
      <a class="navbar-brand" href="/"></a>
    </div>

```

```

</nav>
<div class="container-fluid main ">

  <div class="row container-fluid content">

    <!-- for image Slider -->
    <div class="col-4 img_slider">
      <div>
        
      </div>
    </div>
    <div class="col-4"></div>

    <!-- Form for login with post method -->
    <div class="col-4 Form_area justify-content-center">
      <form action="/login" method="post">

        <div class="logo p-3">
          <h1>Login</h1>
        </div>

        <div class="row">
          <!-- UserName -->
          <div class="form-floating mb-3">
            <input type="text" name="username" class="form-control"
id="floatingInput" placeholder="name@134">
            <label for="floatingInput">UserName</label>
          </div>
        </div>

        <div class="row">
          <!-- Email Address -->
          <div class="form-floating mb-3">
            <input type="email" name="email" class="form-control"
id="floatingInput" placeholder="name@example.com">
            <label for="floatingInput">Email address</label>
          </div>
        </div>

        <!-- password -->
        <div class="row">
          <div class="form-floating mb-3">
            <input type="password" name="password" class="form-control"
id="floatingPassword" placeholder="Password">
            <label for="floatingPassword">Password</label>
          </div>
        </div>
      </form>
    </div>
  </div>

```

```

    <!-- Login / Signup Button -->
    <div class="row">

        <div class="col-6 mb-3">
            <label for="submit">
                <font style="font-size: large; color: brown; font-weight: 600;">*
New User</font>
            </label><br>
            <button type="submit" class="btn btn-primary">Sign in</button>
        </div>

        <div class="col-6 mb-3">
            <label for="submit">
                <font style="font-size: large; color: brown; font-weight: 600;">*
Already a User</font>
            </label><br>
            <a href="/signup.ejs"><button type="submit" class="btn btn-
primary">Sign Up</button></a>
        </div>
    </div>

    </form>
</div>

</div>

</div>

</body>

</html>

```

**Code for the Signup file:**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EduLoop | SignUp</title>
  <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
qKXV1j0HvMUeCBQ+QVp7JcfG1760yU08IQ+GpUo5h1bpg51QRiuqHAJz8+BrxE/N"
crossorigin="anonymous"></script>

  <style>
    nav{
      background-color: rgba(255, 255, 255, 0.35);
      /* background-color: rgb(184, 98, 27,0.5); */
      margin-bottom: 30px;
    }
    body{
      background-image: linear-gradient(#055482, #af70ca);
      background-attachment: fixed;
    }
    .form-floating:not(.form-control:disabled)::before {
      height: 0;
    }

    .Form_area {
      background-color: rgba(255, 255, 255, 0.2);
      padding: 30px;
      border: 1px solid black;
      border-radius: 15px;
    }

    label {
      padding: 5px;
    }
  </style>
</head>

<body>
  <div class="container">
    <div class="row">
      <div class="col-12">
        <div class="text-center">
          <h1>EduLoop</h1>
          <h2>Sign Up</h2>
        </div>
        <div class="Form_area">
          <div class="form-floating">
            <input type="text" class="form-control" value="Name" />
            <label>Name</label>
          </div>
          <div class="form-floating">
            <input type="text" class="form-control" value="Email" />
            <label>Email</label>
          </div>
          <div class="form-floating">
            <input type="password" class="form-control" value="Password" />
            <label>Password</label>
          </div>
          <div class="form-floating">
            <input type="password" class="form-control" value="Confirm Password" />
            <label>Confirm Password</label>
          </div>
          <div class="text-center">
            <button type="button" class="btn btn-primary">Sign Up</button>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        .content {
            margin-top: 10px;
            padding-right: 25px;
        }
        .image_slider{

            box-shadow:5cm rgb(210, 206, 206);
        }
        img{
            margin-left: 10px;
            border-radius: 10px;
        }

</style>

<script>

    var current = 0;

    function nextImg() {
        var x = document.getElementById("image");
        var imgs = ["bg1.jpg", "bg2.jpg", "bg3.jpg",
"bg4.jpg","bg5.jpg","bg6.jpg"];

        if(current<5)
        {
            current=current+1;
        }
        else{current=0;}

        x.src=imgs[current];
    }

    setInterval(nextImg,3500);

</script>
</head>

<body>
    <nav class="navbar navbar-expand-sm">
        <div class="container-fluid">
            <a href="/" class="navbar-brand"></a>
        </div>
    </nav>
    <div class="row container-fluid content">
        <!-- for image Slider -->
        <div class="col-sm-6">

```

```

        <div class="img-slider my-5 py-5">
            
        </div>
    </div>
    <div class="col-sm-1"></div>

    <div class="col-sm-5 Form_area p-5 justify-content-center">
        <!-- form starts here-->
        <form id="form_id">

            <!-- name First and last -->
            <div class="row g-3">
                <div class="col-6 form-floating g-3 mb-3">
                    <input type="text" class="form-control fname"
id="floatingInput" placeholder="Your">
                    <label for="floatingInput">First Name</label>
                </div>
                <div class="col-6 g-3 form-floating mb-3">
                    <input type="text" class="form-control lname"
id="floatingInput" placeholder="Name">
                    <label for="floatingInput"> Last Name</label>
                </div>
            </div>
            <!-- Mobile Number -->
            <div class="row">
                <div class="form-floating mb-3">
                    <input type="tel" class="form-control phno"
id="floatingInput" placeholder="1112223334">
                    <label for="floatingInput"> Mobile No</label>
                </div>
            </div>

            <!-- Address/country/City/state/Zip -->
            <div class="row g-1">
                <!-- address -->
                <div class="col-12 form-floating g-3 mb-3">
                    <input type="text" class="form-control addr"
id="floatingInput" placeholder="Main road 14">
                    <label for="floatingInput">Address</label>
                </div>
            </div>
            <div class="row">
                <!-- country -->
                <div class="col-3 form-floating g-3 mb-3">
                    <input type="text" class="form-control country"
id="floatingInput" placeholder="Country">
                    <label for="floatingInput">Country</label>

```

```

        </div>
        <!-- city -->
        <div class=" col-3 form-floating g-3 mb-3">
            <input type="text" class="form-control city"
id="floatingInput" placeholder="City">
            <label for="floatingInput">City</label>
        </div>
        <!-- state -->
        <div class="form-floting col-md-4 mb-3">

            <select class="form-select state" id="floatingSelect" aria-
label="States">

                <option value="Andhra Pradesh">Andhra Pradesh</option>
                <option value="Arunachal Pradesh">Arunachal
Pradesh</option>

                <option value="Assam">Assam</option>
                <option value="Bihar">Bihar</option>
                <option value="Chhattisgarh">Chhattisgarh</option>
                <option value="Goa">Goa</option>
                <option value="Gujarat">Gujarat</option>
                <option value="Haryana">Haryana</option>
                <option value="Himachal Pradesh">Himachal
Pradesh</option>

                <option value="Jharkhand">Jharkhand</option>
                <option value="Karnataka">Karnataka</option>
                <option value="Kerala">Kerala</option>
                <option value="Madhya Pradesh">Madhya Pradesh</option>
                <option value="Maharashtra">Maharashtra</option>
                <option value="Manipur">Manipur</option>
                <option value="Meghalaya">Meghalaya</option>
                <option value="Mizoram">Mizoram</option>
                <option value="Nagaland">Nagaland</option>
                <option value="Odisha">Odisha</option>
                <option value="Punjab">Punjab</option>
                <option value="Rajasthan">Rajasthan</option>
                <option value="Sikkim">Sikkim</option>
                <option value="Tamil Nadu">Tamil Nadu</option>
                <option value="Telangana">Telangana</option>
                <option value="Tripura">Tripura</option>
                <option value="Uttar Pradesh">Uttar Pradesh</option>
                <option value="Uttarakhand">Uttarakhand</option>
                <option value="West Bengal">West Bengal</option>
            </select>
        </div>
        <!-- zip -->
        <div class=" col-2 form-floating g-3 mb-3">

```

```

        <input type="tel" class="form-control pincode"
id="floatingInput" placeholder="111111">
        <label for="floatingInput">Zip</label>
    </div>
</div>
<!-- username -->
<div class="row">
    <div class="col-6 form-floating g-3 mb-3">
        <input type="text" class="form-control fname"
id="floatingInput" placeholder="Username">
        <label for="floatingInput">UserName</label>
    </div>
</div>
<div class="row">
    <!-- Email Address -->
    <div class="form-floating mb-3">
        <input type="email" class="form-control p-3 email"
id="floatingInput" placeholder="name@example.com">
        <label for="floatingInput">Email address</label>
    </div>
</div>

<!-- password -->
<div class="row">
    <!-- first time -->
    <div class="col-6 form-floating mb-3">
        <input type="password" class="form-control pswrd"
id="floatingPassword" placeholder="Password">
        <label for="floatingPassword"> Password</label>
    </div>
    <!-- re-entering -->
    <div class="col-6 form-floating mb-3">
        <input type="password" class="form-control cpswrd"
id="floatingPassword" placeholder="Password">
        <label for="floatingPassword">Re-enter Password</label>
    </div>
</div>

<!-- asking for Advertising -->
<div class="form-check mb-3">
    <input class="form-check-input" type="checkbox" id="gridCheck">
    <label class="form-check-label" for="gridCheck">
        Send me updates via Email*
    </label>
</div>

<!-- Login / Signup Button -->
<!-- use Stacks in this from bootstrap -->

```



```

        <div class="row">

            <div class="col-6 mb-3">
                <label for="submit">
                    <font style="font-size: large; color: brown; font-
weight: 600;">* New User</font>
                </label><br>
                <button type="submit" class="btn btn-primary"
onclick="validate_form()">Sign Up</button>
            </div>

            <div class="col-6 mb-3">
                <label for="submit">
                    <font style="font-size: large; color: brown; font-
weight: 600;">* Already a User</font>
                </label><br>
                <a href="login.html"><button type="submit" class="btn btn-
primary">Sign in</button></a>
            </div>
        </div>
    </div>
</div>

</form>
</div>
</div>

<script>

function validate_form()
{
    function validating_form(){
        var first_name=document.getElementsByClassName('fname')[0];
        var last_name=document.getElementsByClassName('lname')[0];
        var phone_no=document.getElementsByClassName('phno')[0];
        var email_id=document.getElementsByClassName('email')[0];
        var username = document.getElementsByClassName('fname')[1];
        var password = document.getElementsByClassName('pswrd')[0];
        var c_password = document.getElementsByClassName('cpswrd')[0];

        if(namevalidation(first_name,last_name))
        {
            if(phone_validation(phone_no))
            {
                if(email_validation(email_id))
                {
                    if(password_validation(password,c_password))
                    {
                        return true;

```

```

    }
    }
}
else{
    return false;
}

function namevalidation(name1,name2)
{
    var val1=name1.value;
    var val2=name2.value;

    if( (val1.length>12) || (val1.length<3) )
    {
        alert('First Name must be 3 to 12 characters long!');
        return false;
    }
    else if( (val2.length>12) || (val2.length<3) )
    {
        alert('Last Name must be 3 to 12 characters long!');
        return false;
    }
    else{
        return true;
    }
}

function phone_validation(phone_no)
{
    var num_regex=/^[0-9]{10}$/;

    var len=phone_no.value.length;

    if(phone_no.value.match(num_regex))
    {
        return true;
    }
    else{
        if(len == 0)
        {
            alert('Phone number cannot be empty!');
            return false;
        }
        alert('Phone number must be in 10 digits');
        return false;
    }
}

```

```

        function email_validation(email)
        {
            var email_regex=/^([a-zA-Z0-9_\.\-]+\@(([a-zA-Z0-9\-\-])+\.))+([a-zA-Z0-9]{2,4})+$/;
            if(email.value.match(email_regex))
            {
                return true;
            }
            else{
                alert("Please enter Valid Email Address !");
                email.focus();
                return false;
            }
        }
        function password_validation(pass,cpass)
        {
            if(pass.value.length < 8 || cpass.value.length < 8)
            {
                alert('Password and/or Confirm password field cannot be empty
!\nAnd should atleast 8 charcters long');
                return false;
            }
            if(pass.value != cpass.value)
            {
                alert("Password and Confirm Password Must be Same");
                return false;
            }else{
                return true;
            }
        }
        }
        console.log(validating_form());
        if(validating_form())
        {
            console.log('entered if statement')
            var form_attr= document.getElementById('form_id');
            form_attr.setAttribute('action','/sign_up');
            form_attr.setAttribute('method','post');
            console.log(form_attr.getAttribute('action'));
            console.log(form_attr.getAttribute('method'));
        }
    }
</script>

</body>
</html>

```

---

*END OF FILE*

---

**THANK YOU**