User_registration form

- **Team Member details:**
- 1:Mohammad Saif :
- Collage->United Collage of Engineering(UIT)
- University_rollnu->2002840100079
- E:mail->mdsaifcs1023@gamil.com
- Mob: ->87390468866

- 2:Shubham Singh:
- Collage->United Collage of Engineering and Research(UCER)
- University_rollnu->2100101530056

- 3:Abhinav Tripathi
- University_rollnu->2100101530006
- Collage->United Collage of Engineering and Research(UCER)

Technology Used:

- Front_End:
- 1:HTML
- 2:CSS
- 3:JAVASCRIPT

- Back_End:
- 1:NODE_JS
- 2:MONGO_DB

DESCRIPTION OF PROGARMMING LANGUAGE AND TECHNOLOGIES:

- PART 1:FRONT_END:
- 1:HTML:
- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
- 2:CSS:
- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- 3:JAVASCRIPT:
- **JavaScript** is a lightweight, interpreted **programming** language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

DESCRIPTION OF PROGARMMING LANGUAGE AND TECHNOLOGIES:

- **PART 2:BACK_END:**

- **1:NODE_JS:**
- Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project!
- Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.
- A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

- **2:MONGO_DB:**

- MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

MODULES ,FILES USED

- FRONT_END:
-
- 1:Register.html
- 2:App.js
- 3:Style.css

- BACK_END:
-

1:Database:

1:homeSchema.js
2:mongoose module

2:Node_js:

1:homeRouter.js
2:express module

Complimetry:

package.json

MODULES ,FILES USED DESCRIPTION

- PART 1:
- FRONT_END:
- **Register.ejs**

- **this file is used for making the data visible on web page .**
- **This file also involves the user form where the user can fill the data according to the text label**

and can put the valid data.
Using post method in form we can data form
the user and through the server we can put it
into our database

App.js file

We will be treating this file as a serevr creater file.
we will create server (node server module) express to make communtication
with front end and backend(database)

MODULES ,FILES USED DESCRIPTION

- Back_End:

- 1:homeschema.js

  In this file we will create a database schema to get and store the data intdatabase:

2:homerouter.js

In this file we will creating router for cheacking and validating the data that we will be getting from user

Register.ejs

```
<!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>mohammad saif</title>
    <style>
        body,html{
            height: 100%;
            background-color: rgb(44, 44, 82);
        }
     h1{
        text-align: center;
        color:white;
     }
        #back{
           background-image:url("img/img1a.webp");
           height: 100%;
           background-position: center;
           background-repeat: no-repeat;
           background-size: cover;
        }
        ul{
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
        li{
            float: left;
            display: block;
            color: white;
            text-align: center;
            padding: 14px 16px;
            text-decoration: none;
            font-weight: 30px;
        }
        footer{
            background-color:white;
            display: grid;
            height: 60px;
            color: rgb(12, 12, 99);
        }

        .row {
  display: -webkit-flex;
  display: flex;
}

/* Create three equal columns that sits next to each other */
.column{
  -webkit-flex: 1;
  -ms-flex: 1;
  flex: 1;
  padding: 20px;
  padding-left: 20px;
  height: 250px; /* Should be removed. Only for demonstration */
  background-color: rgb(114, 113, 113);
  color: white;
  font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
  font-size: 20px;


}
@media (max-width: 600px) {
  .row {
    -webkit-flex-direction: column;
```

```
@media (max-width: 600px) {
  .row {
    -webkit-flex-direction: column;
    flex-direction: column;
  }
}
.cen{
     color: white;
     font-size: 20px;
     font-family: Arial, Helvetica, sans-serif;
     align-content: center;
}

.jk{
 text-align: center;
 font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
 font-style: normal;
 font-size: 25px;
 font-weight: bold;
 color: white;
}
form{
    margin-left: 500px;
    margin-right: 500px;
}
fieldset{
    padding: 20px;
    background-color: white;
    border-radius: 20px;
    border-width: 7px;
    border-color: black;

}
input{
    font-size: 20px;
    font-family: Georgia, 'Times New Roman', Times, serif;
    border-radius: 20px;
    background-color: rgb(240, 196, 203);
    border-color:white;
    border-width: 9px;
    border-block-color: black;
    padding: 5px;
}
input:hover{
    background-color: white;
}

label{
     font-family: 20px;
     font-size: 20px;
    font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
}
button{
    padding: 10px;
    background-color:rgb(172, 230, 184);
    color: black;
    font-weight: 10px;
    border-radius: 5px;
}
button:hover{
  background-color:rgb(244, 245, 244);
  cursor: pointer;
  color: black;
  font-weight: 10px;
}
hr{
    color: white;

}

.im{
    border-radius: 25px;
    width: 40px;
    background-color: black;
```

```
hr{
    color: white;

}

.im{
    border-radius: 25px;
    width: 40px;
    background-color: black;
    padding: 5px;
    margin-top: 5px;
}
.mob{
    display: none;}

            .mob{
                display: inline;
            }


    </style>
</head>
<body>
    <div class="mob">
        <ul>
        <li>Home</li>
        <li>Contact</li>
        <li>News</li>
        <li>About</li>
        <li>Logout</li>
         <br>
         <hr style="color:white;">
        <br>
        <div class="jk">
            <h2 id="hp">Welcome</h2>
            <p id="jk">you can register here to make Contact with us</p>
            <p id="jk">we are here to solve your problem:</p>
            <p id="jk">you have to log in first</p>
        </div>
    </ul>

    <div id="back">
        <h1><%=title %> </h1>
        <h1><%=password %> </h1>
        <h1><%=email %> </h1>
       <form action="/register" method="post">
        <fieldset>
           username <br>
           <input type="text" name="username" id="user"><br>
           phone_number <br>
           <input type="text" name="phone_number" id="pn"><br>
           email <br>
           <input type=" email" name="email" id="em"><br>
           password <br>
           <input type="password" name="password" id="pas">
           <br>
           conform_password <br>
           <input type="password" name="conform_password" id="pas"><br>
           <br>
           <button type="submit">submit</button>
       </form>
</div>
       <footer>
        <div class="row">
            <hr>
  <div class="column" style="background-color:black;">Sign-up With

   <h4>Google</h4>
   <h4>Facebook</h4>
   <h4>Twiter</h4>

</div>
  <div class="column" style="background-color:black;">CONTACT
  <h3>Gmail:</h3>
```

```
<div class="row">
        <hr>
 <div class="column" style="background-color:black;">Sign-up With

  <h4>Google</h4>
  <h4>Facebook</h4>
  <h4>Twiter</h4>

</div>
 <div class="column" style="background-color:black;">CONTACT
 <h3>Gmail:</h3>
 <h4>mdsaifcs1023@gmail.com</h4>
 <h3>mob:</h3>
 <h4>8739048866</h4>
</div>
 <div class="column" style="background-color:black;">OWNER</div>
     </div>
   </footer>
</div>
</body>
</html>
```

App.js

```
const express=require('express');
const mongoose=require('mongoose');
const bodyParser=require('body-parser');
const homeRouter=require('./routers/homeRouter');

const port=process.env.port||3000;



const app=express();

// db connection
mongoose.connect('mongodb://localhost:27017/studentsdata3',{useNewUrlParser:true})
const db=mongoose.connection;

db.on("error",()=>{
    console.log("error in connection ");
})
db.once('open',()=>{
    console.log("connected");
})

app.set('view engine','ejs')

app.use(express.static('public'))

/*

app.get('/',(err,res)=>{
    res.send("hello");
})
*/



app.use(bodyParser.urlencoded({extended:false}))
app.use(bodyParser.json())



app.use('/',homeRouter)



app.listen(port)
```

homeSchema.js

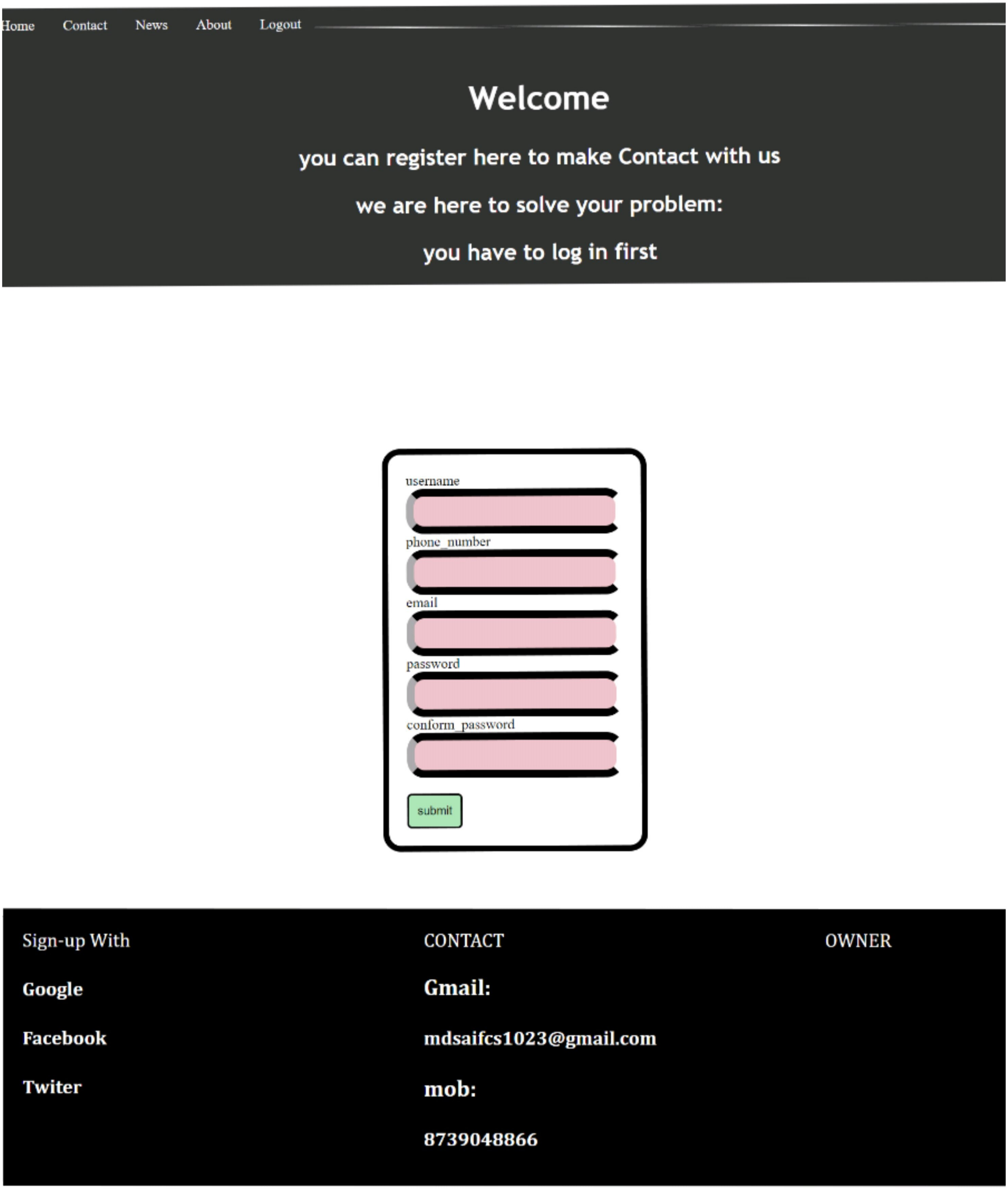
```
const mongoose=require('mongoose');

const schema=mongoose.Schema;
const userSchema=new schema({
    username:{
        type:String,
        required:true
    },
    phone_number:{
        type:Number,
        required:true
    },
    email:{
        type:String,
        required:true,
        unique:true
    },
    password:{
        type:String,
        required:true,
        unique:true
    }
})
module.exports=mongoose.model('Registerusers3',userSchema)
```

homeRouter.js

```
const express=require('express');
const Router=express.Router();
const homeSchema=require('../models/homeSchema')

Router.get('/',(err,res)=>{
res.render('register',{title:'fill-form',password:'',email:''})
})

Router.post('/register',async(req,res)=>{
try{
 const {
    username,
    phone_number,
    email,
    password,
    conform_password
 }=req.body;

 if(password===conform_password){
       const userData=new homeSchema({
        username,
        phone_number,
        email,
        password
       })
       userData.save(err=>{
        if(err){
        res.render('register',{title:'error',password:'',email:''})
        }

        else{
         res.render('register',{title:'Done',password:'',email:''})
        }

       })
 }


 else{
res.render('register',{title:' ',password:'password not match',email:''})
 }
}

catch(error){

    res.render('register',{title:'error',password:'',email:''})
}
})
module.exports=Router;
```

package.json

```json
{
  "name": "backpro3",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ejs": "^3.1.8",
    "express": "^4.18.1",
    "mongodb": "^4.9.0",
    "mongoose": "^6.5.4"
  }
}
```



Welcome

you can register here to make Contact with us

we are here to solve your problem:

you have to log in first

Sign-up With

Google

Facebook

Twiter

CONTACT

Gmail:

mdsaifcs1023@gmail.com

mob:

8739048866

OWNER

Connect
View
Collection
Help
localhost:27017
8 DBS
8 COLLECTIONS
FAVORITE
My Queries
Databases
Filter your data
local
studentsdata
studentsdata1
studentsdata3
registerusers3
>_MONGOSH
Documents
studentsdata3.regi...
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
FILTER
{ field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Type here to search
32°C Rain showers
ENG
IN
9:52 AM
9/3/2022

MongoDB Compass - localhost:27017/studentsdata3.registerusers3

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 6.0.1 Community

studentsdata3.registerusers3

3
DOCUMENTS

Displaying documents 1 - 3 of 3

```
_id: ObjectId('630f00f4aa5be5698df42797')
username: "mdsaif_1007"
phone_number: 8739048866
email: "mdsaifcs@gmail.com"
password: "12"
__v: 0

_id: ObjectId('630f11f62eb0ffdf37392582')
username: "1234"
phone_number: 8739048866
email: "sfg@google.com"
password: "1234"
__v: 0

_id: ObjectId('6310ca8b54ff101c21e0b0d9')
```