

Week2 :design a web page using bootstrap tempelet.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Week2 Bootstrap</title>

<style>

footer {

color: white;

padding: 15px;

bottom: 0;

position: absolute;

}

</style>

</script>

</head>

<body>

<nav class="navbar navbar-expand navbar-dark bg-success">

<div class="container-fluid">

tags

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

About us

```
<li class="nav-item">
<a class="nav-link" href="">Contact</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle"
role="button"
id="navbarDropdown"
data-bs-toggle="dropdown"
href="">Episodes</a>
<ul class="dropdown-menu">
<li>
<a class="dropdown-item" href="">Episode1</a>
</li>
<li>
<a class="dropdown-item" href="">Episode2</a>
</li>
</ul>
</li>
</ul>
<form class="d-flex">
<input class="form-control me-2"
type="search"
placeholder="search website"/>
<button class="btn btn-primary"
type="submit">search</button>
</button>
</form>
</div>
```

```
</nav>

<div class="container mt-5">

<div class="row">

<div class="col-sm-4">

<h3>Doraemon Biography</h3>

</div>

<div class="col-sm-8">

<h2>Article Title</h2>

<h5>Created on 06 march 2024</h5>

<p>Doraemon is a male robotic earless cat that travels back in time from
the

22nd century to aid a preteen boy named Nobita.

An official birth certificate for the character gives him a birth date of 3
September 2112 and lists his city of residency as Kawasaki, Kanagawa,
the city where the manga was created.</p>

</div>

</div>

</div>

<footer class="container-fluid text-center bg-success">

<p>copy right CMRIT_2024<br>

</footer>

</body>

</html>
```

Week3:wfrite a java script code to valadite user rigistation and and login form.

Registration.html

<!DOCTYPE html>

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Registration Page</title>
//css Design part
<style>
fieldset{
width:500px;
background-image:url(n1.jpg);
background-repeat:no-repeat;
background-size:cover;
}
table tr td{
padding-top:6px;
padding-bottom:6px;
}
body{
color:white;
}
</style>
//validation using java script code
<script>
function validate(){
let fn=frm.fname.value;
for(let x in fn){
let ch=fn.charCodeAt(x);
```

```
if(ch<65||ch>90 && ch<97||ch>122)
{
  alert("Invalid firstname");
  return false;
}
}

let fn1=frm.lname.value;
for(let x in fn1){
  let ch=fn1.charCodeAt(x);
  if(ch<65||ch>90 && ch<97||ch>122)
  {
    alert("Invalid Lastname");
    return false;
  }
}

let phn=frm.phone.value;
let len=phn.length;
if(len!==10)
{
  alert("phone no. should be exactly 10 digits");
  return false;
}

let reg="@";
let mail=frm.email.value;
if(reg.test(mail))
{
  alert("Invalid mail");
  return false;
}
```

```

}
let p1=frm.pwd.value;
let p2=p1.length;
if(p2>8)
{
alert("Password should not exceed 8 digits");
return false;
}
return true;
}
</script>
</head>

//creating registration form

<body style="background-image:url(nature.jpg);background-
repeat:no-repeat;background-size:cover">

<center>

<h2>Registration</h2>

<form name="frm" method="post" action="success.html"
onSubmit="return validate()">

<fieldset>

<table>

<tr><td>First Name:</td><td><input type="text"
name="fname" value="" required/> </td></tr>

<tr><td>Last Name:</td><td><input type="text"
name="lname" value="" required/> </td></tr>

<!--<label for="num">Phone No.:</label>

<input type="tel" name="Mobile" id="num" maxlength="10"
placeholder="Enter user Mobile No." required />-->

```

```
<tr><td>Phone no.:</td><td><input type="number"
name="phone" value="" required/> </td></tr>
<tr><td>Mail Id:</td><td><input type="email"
name="mailid" value="" required/> </td></tr>
<tr><td>Gender:</td>
<td>Male:<input type="radio" name="gender"
value="male"/> </td>
<td>Female:<input type="radio" name="gender"
value="female"/> </td>
</tr>
<tr><td>DOB:</td><td><input type="date" name="dob"
required/> </td></tr>
<tr><td>User Name:</td><td><input type="text"
name="uname" value="" required/> </td></tr>
<tr><td>Password:</td><td><input type="password"
name="pwd" value="" required/> </td></tr>
<tr><td>Age:</td><td><input type="number" name="age"
value="" required/> </td></tr>
</table>
<input type="submit" value="Registration"
name="submit"/>
//login button if existing user
<a href="login.html"><input type="button" value="login"
name="submit"/></a>
</fieldset>
<p>Copyright CMRIT_2024</p>
</form>
</center>
```

</body>

</html>

Code:-Success.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

**<meta name="viewport" content="width=device-width,
initial-scale=1.0">**

<title>Document</title>

</head>

<body>

<h2>Registration Successfull</h2>

</body>

</html>

Code:-login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

**<meta name="viewport" content="width=device-width, initial-
scale=1.0">**

<title>Document</title>

//css Design part

<style>

fieldset{

width:500px;

background-image:url(nature.jpg);


```
background-repeat:no-repeat;
background-size:cover;
}
table tr td{
padding-top:6px;
padding-bottom:6px;
}
</style>

//validation using java script code
<script>
function validate(){
let p1=frm.pwd.value;
let p2=p1.length;
if(p2>8)
{
alert("Password should not exceed 8 digits");
return false;
}
return true;
}
</script>
</head>

//creating login form
<body style="background-image:url(n1.jpg);background-repeat:no-
repeat;background-size:cover">
<center>
<h2>Login</h2>
<form name="frm" method="post" action="abc.html" onSubmit="return
```

```

validate()">
<fieldset>
<table>
<tr><td>User Name:</td><td><input type="text" name="uname"
value="" required/> </td></tr>
<tr><td>Password:</td><td><input type="password" name="pwd"
value="" required/> </td></tr>
</table>
<input type="submit" value="Login" name="submit"/>
//registration button if new user
<a href="Registration.html"><input type="button" value="Registration"
name="submit"/></a>
</fieldset>
<p>Copyright CMRIT_2024</p>
</form>
</center>
</body>
</html>
Code:-Success2.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Document</title>
</head>
<body>

```

<h2>Login Successfull</h2>

</body>

</html>

Week4: Write a jquery code to show the website slider.

<!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>jQuery</title>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script

src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<div id="myCarousel" class="carousel slide">

<ol class="carousel-indicators">

<li class="item1 active">

<li class="item2">

<li class="item3">

```
</ol>

<div class="carousel-inner" role="listitem">

  <div class="item active">

    

  </div>

  <div class="item">

    

  </div>

  <div class="item">

    

  </div>

</div>

<a class="left carousel-control" href="#myCarousel" role="button">

  <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>

  <span class="sr-only">Prev</span>

</a>

<a class="right carousel-control" href="#myCarousel" role="button">

  <span class="glyphicon glyphicon-chevron-right"
  aria-hidden="true"></span>

  <span class="sr-only">Next</span>

</a>

</div>

</div>

<script>

$(document).ready(function(){

  $("#myCarousel").carousel();

  $(".item1").click(function(){

    $("#myCarousel").carousel(0);
```

```
})  
$(".item2").click(function(){  
  $("#myCarousel").carousel(1);  
})  
$(".item3").click(function(){  
  $("#myCarousel").carousel(2);  
})  
$(".left").click(function(){  
  $("#myCarousel").carousel("prev");  
})  
$(".right").click(function(){  
  $("#myCarousel").carousel("next");  
})  
});  
</script>  
</body>  
</html>
```

Week5: Create a simple Node.js server with routes login, register, profile and logout.

index.html:-

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title>Week 3 Register and Login page</title>  
<style>  
a{
```



```

}

table tr td {
padding-top: 6px;
padding-bottom: 6px;
}

fieldset{width:500px; text-align: center;margin: auto;}

</style>

<div id="header">

<h1>Registration</h1>

</div>

<div id="content">

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Register</title>

</head>

<script>

function validate()

{

var fn = frm.fname.value;

localStorage.setItem("firstname",fn)

for(x in fn)

{

ch=fn.charCodeAt(x);

if(ch<65||ch>90 && ch<97||ch>122)

{

alert("Invalid firstname");

return false;

}

}

}

```

```
var ln = frm.lname.value;  
localStorage.setItem("lastname",ln)  
for(y in ln)  
{  
ch=ln.charCodeAt(y);  
if(ch<65||ch>90 && ch<97||ch>122)  
{  
alert("Invalid lastname");  
return false;  
}  
}  
var phn = frm.phone.value;  
localStorage.setItem("phone",phn)  
var lenp = phn.length;  
if (lenp!=10)  
{ alert("Phone no should be exactly 10 digits");  
return false;  
}  
var pwd1=frm.pwd.value;  
localStorage.setItem("password",pwd1)  
var pwdl=pwd1.length;  
if(pwdl%2===1)  
{  
alert("Password should contain even number of characters");  
return false;  
}  
if(pwdl > 8)  
{
```



```
alert("Password should not exceed 8 digits");
return false;
}
var reg = /^[a-z0-9]+@[a-z]+\.[a-z]{2,3}$/;
var mail=frm.mailid.value;
localStorage.setItem("email",mail)
if (reg.test(mail)){
alert("Registration Successful!!");
}
else{
alert("Invalid email");
return false;
}
return true;
}
</script>
<div id="container">
<body>
<center>
<form name="frm" method="POST" action="/login" onSubmit="return
validate()">
<fieldset align="center">
<table align="center">
<tr><td>First Name: </td><td><input type="text"
name="fname" value="" size="50" required/></td></tr>
<tr><td>Last Name: </td><td> <input type="text" name="lname"
value="" size="50" required/></td></tr>
<tr><td>Phone No: </td><td> <input type="text" name="phone"
```

```

value="" size="50" required/></td></tr>
<tr><td>Mail id:</td><td><input type="email" name="mailid"
value="" size="50" required/></td></tr>
<tr><td>Gender:</td>
<td>Male: <input type="radio" name="gender" value="male" >
Female: <input type="radio" name="gender"
value="female"></td></tr>
<tr><td>DOB :</td><td><input type="date" name="dob" size="50"
required/></td></tr>
<tr><td>Username:</td><td><input type="text" name="uname"
value="" size="50" required/></td></tr>
<tr><td>Password:</td><td><input type="password" name="pwd"
value="" size="50" required/></td></tr>
<tr><td>Age:</td><td><input type="text" name="age" value=""
size="50" required/></td></tr>
</table>
<input type="submit" value="SUBMIT" name="submit" />
</fieldset>
</form>
</div>
</center>
</body>
<div id="footer">
Copyright © CMRIT_2024
</div>
</div>
</html>

```

login.html:-

```
<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

</head>

<script>

function LoginValidate(){

var enteremail=document.getElementById('email').value;

var enterpassword=document.getElementById('pwd').value;

var getEmail=localStorage.getItem('email')

var getPwd=localStorage.getItem('password')

if(enteremail==getEmail)

{

if(enterpassword==getPwd)

{

alert("login success");

window.location= "/welcome" ;

return false;

}

else

{

alert("wrong password");

}

}

else

{

alert("invalid details");

}

}
```

```
}  
</script>  
<body style="text-align: center;">  
<div class="main">  
<h1>Login Form</h1>  
<h3>Enter your login credentials</h3>  
<form >  
<label for="first">  
Email:  
</label>  
<input type="text" name="first" id="email"  
placeholder="Enter your Username" required>  
<br> <br>  
<label for="password">  
Password:  
</label>  
<input type="password" name="password" id="pwd"  
placeholder="Enter your Password" required>  
<br> <br>  
<button type="submit" onclick="return LoginValidate()">Submit</button>  
<br>  
</div>  
</form>  
<p>Not registered?  
<a href="/register"  
style="text-decoration: none;">  
Create an account  
</a>
```

</p>

</div>

</body>

</html>

welcome.html:-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>welcome</title>

</head>

<script>

function LogoutValidate(){

localStorage.clear();

}

function getdetails(){

var getfirstname=localStorage.getItem("firstname") ;

var getemail=localStorage.getItem("email")

document.getElementById("data").innerHTML="welcome to

**" +getfirstname+"
" +getemail;**

}

</script>

<body onload="getdetails()">

<h1 id="data">Welcome to Cmrit</h1>

<form action="/" method="post">

<button type="submit" onclick="LogoutValidate()">Logout</button>

</form>

</body>

</html>

server.js:-

// Importing express module

const express = require('express');

const app = express();

//app.use(express.json());

app.get('/',

(req, res) => {

res.sendFile(__dirname + '/index.html');

});

app.get('/login',

(req, res) => {

res.sendFile(__dirname + '/login.html');

});

app.get('/register',

(req, res) => {

res.sendFile(__dirname + '/register.html');

});

app.get('/welcome',

(req, res) => {

res.sendFile(__dirname + '/welcome.html');

});

app.post('/',(req, res) => {

res.redirect("/")

});

app.post('/login',(req, res) => {

res.redirect("/login")

```

});
app.post('/register',(req, res) => {
res.redirect("/register")
});
app.post('/welcome',(req, res) => {
res.redirect("/welcome")
});
app.listen(3000,
() => {
console.log(
'Our express server is up on port 3000'
);
});

```

Week6: Write middleware to validate users and generate JWT.

database.json:-

```

[ {"name":"anusha","work":"Assistant Professor","password":"anu"},
{"name":"radha","work":"Assistant Professor","password":"radha12"},
{"name":"jaya","work":"Assistant Professor","password":"jaya12"}
]

```

login.html:-

```

<!DOCTYPE html>
<html>
<head>
<title>Login</title>
<style>
h2{
display:none;

```

```
}  
  
</style>  
  
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  
  
<script>  
  
const form=$("#loginForm");  
  
$(document).ready(function(){  
  
$("#loginForm").submit(  
  
(event)=>{  
  
event.preventDefault();  
  
var name=$("#username").val();  
  
var password=$("#pwd").val();  
  
$.ajax({  
  
url:"/auth",  
  
type:"POST",  
  
contentType:"application/json",  
  
dataType:"json",  
  
data:JSON.stringify({  
  
name:name,  
  
password:password  
  
})),  
  
success:function(data){  
  
var tokenData=data;  
  
if(tokenData.login==true){  
  
if(verifyLogin(tokenData.token)){  
  
$("#verified").show();  
  
$("#not-verified").hide();  
  
alert("JWT Token: "+tokenData.token);  
  
}  
  
}
```



```
}  
else{  
    $("#verified").hide();  
    $("#not-verified").show();  
    alert("Authentication Failed");  
}  
},  
error:function(data){  
    console.log("Something went wrong");  
}  
});  
});  
});  
function verifyLogin(token)  
{  
    let result=true;  
    $.ajax({  
        url:"/verifyToken",  
        type:"POST",  
        contentType:"application/json",  
        dataType:"json",  
        data:JSON.stringify({  
            token:token,  
        }},  
        success:function(data){  
            if(data.login==true)  
            {  
                result=true;
```

```
}  
else{  
    result=false;  
}  
},  
error:function(data){  
    console.log("wrong Token, Not Authenticated.");  
}  
});  
return result;  
}  
</script>  
</head>  
<body>  
    <center>  
        <h1>Login Page</h1>  
        <form id="loginForm">  
            <label for="username">UserName:</label><br>  
            <input type="text" id="username" name="username"><br>  
            <label for="pwd">Password:</label><br>  
            <input type="password" id="pwd" name="pwd"><br>  
            <input type="submit" value="Submit">  
        </form>  
        <h2 id="verified">You are a verified user</h2>  
        <h2 id="not-verified">You are not a verified user</h2>  
    </center>  
</body>  
</html>
```

server1.js:-

//import express for creating APIs endpoints

const express=require("express");

const path=require("path");

const fs=require("fs");

const users=require("./database.json");

var database;

var token="wrong key";

//Read database.json file

fs.readFile("database.json",function(err,data){

//check for errors

if(err) throw err;

//Converting to JSON

database=JSON.parse(data);

});

const jwt=require("jsonwebtoken");

const app=express();

const port=3001;

app.use(express.json());

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

res.sendFile(__dirname+'/login.html');

});

app.post("/auth",(req,res)=>{

const name=req.body.name;

console.log(name);

const password=req.body.password;

console.log(password);

let isPresent=false;

```
let isPresentIndex=null;
for(let i=0;i<database.length;i++){
  if(database[i].name===name && database[i].password===password){
    isPresent=true;
    isPresentIndex=i;
    break;
  }
}
if(isPresent){
  const token=jwt.sign(database[isPresentIndex],"secret");
  res.json({
    login:true,
    token:token,
    data:database[isPresentIndex],
  });
}
else{
  res.json({
    login:false,
    token:token,
    error:"please check name and password"
  });
}
});
app.post("/verifyToken", (req, res)=>{
  const token=req.body.token;
  if(token){
    const decode=jwt.verify(token,"secret");
```

```
res.json({
  login:true,
  data:decode
});
}
else{
  res.json({
    login:false,
    data:"error"
  });
}
});
app.post('/login',(req, res)=>{
  res.redirect("/login");
});
app.listen(port, ()=>{
  console.log(` Server is running:http://localhost:${
```

week7: Write middleware to validate JWT and redirect to profile

Database. Json:

```
[
  {"name":"jyotsna","work":"Assistant Professor","password":"abc"},
  {"name":"Sarvesh","work":"Assistant Professor","password":"ambala"},
  {"name":"ram","work":"Assistant Professor","password":"priya"},
  {"name":"Lakshmi","work":"Asst Prof","password":"Cmrit@123"}
]
```

Server.js:

```
const express = require("express");
const path = require('path');
const fs = require("fs");
const users = require("../database.json");
const jwt = require("jsonwebtoken");
const app = express();
const port = 3000;
app.use(express.json());
app.get('/',
(req, res) => {
res.sendFile(__dirname + '/index.html');
});
app.get('/welcome',
(req, res) => {
res.sendFile(__dirname + '/welcome.html');
});
app.get('/signup',
(req, res) => {
res.sendFile(__dirname + '/signup.html');
});
app.get('/login',
(req, res) => {
res.sendFile(__dirname + '/login.html');
});
var database;
var token;
app.post("/register", (req, res) => {
const name = req.body.name;
```

```
const password = req.body.password;  
const work = req.body.work;  
let isPresent = false;  
let isPresentIndex = null;  
fs.readFile("database.json", function(err, data) {  
  if (err) throw err;  
  database = JSON.parse(data);  
  for (let i = 0; i < database.length; i++) {  
    if (database[i].name === name  
      && database[i].password === password) {  
      isPresent = true;  
      isPresentIndex = i;  
      break;  
    }  
  }  
// If isPresent is true, then create a  
// token and pass to the response  
if (isPresent) {  
  res.json({  
    signup: false,  
    token: null,  
    error: "Already registered",  
  });  
  } else {  
    let user =  
    {  
      name: name,  
      work: work,
```

```
password: password,
token: token
};
users.push(user);
fs.writeFile(
"database.json",
JSON.stringify(users),
err => {
// Checking for errors
if (err) throw err;
res.json({
signup: true,
token: "generated",
data: "Successfully Registered",
});
console.log("Done writing");
});
}
});
});
app.post("/auth", (req, res) => {
const name = req.body.name;
console.log(name);
const password = req.body.password;
console.log(password)
let isPresent = false;
let isPresentIndex = null;
fs.readFile("database.json", function(err, data) {
```



```
if (err) throw err;

database = JSON.parse(data);

for (let i = 0; i < database.length; i++) {

  if (database[i].name === name

    && database[i].password === password) {

    isPresent = true;

    isPresentIndex = i;

    break;

  }

}

if (isPresent) {

  const token = jwt.sign(database[isPresentIndex], "secret");

  res.json({

    login: true,

    token: token,

    data: database[isPresentIndex],

  });

} else {

  // If isPresent is false return the error

  res.json({

    login: false,

    error: "please check name and password.",

  });

}

});

});

app.post("/verifyToken", (req, res) => {

  const token = req.body.token;
```

```
if (token) {  
  const decode = jwt.verify(token, "secret");  
  res.json({  
    login: true,  
    data: decode,  
  });  
} else {  
  // Return response with error  
  res.json({  
    login: false,  
    data: "error",  
  });  
}  
});  
  
app.post('/welcome',(req, res) => {  
  res.redirect("/welcome")  
});  
  
app.post('/login',(req, res) => {  
  res.redirect("/login")  
});  
  
app.post('/signup',(req, res) => {  
  res.redirect("/signup")  
});  
  
// Listen the server  
app.listen(port, () => {  
  console.log(`Server is running :  
http://localhost:${port}/`);  
});
```

Login.html:

<!DOCTYPE html>

<html>

<head>

<title>Login Page</title>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>

const form=\$("#loginForm");

\$(document).ready(function() {

\$("#loginForm").submit((event)=>{

event.preventDefault();

var name = \$("#username").val();

var password=\$("#pwd").val();

//generateToken by making a service call to /auth with post

\$.ajax({

url:"/auth",

type:"POST",

contentType:"application/json",

dataType:"json",

data: JSON.stringify({

name: name,

password: password}),

success: function(data) {

var tokenData=data;

if(tokenData.login==true){

localStorage.setItem("username",tokenData.data.name);

localStorage.setItem("job",tokenData.data.work);

if(verifyLogin(tokenData.token))

```
{  
  window.location.href="/welcome";  
}  
else{  
  alert("Authentication Failed");  
}  
}  
},  
error: function(data ){  
  console.log("Something went wrong");  
}  
});  
});  
});  
function verifyLogin(token)  
{  
  let result=true;  
  //verifyToken by making a service call to /verifyToken with get  
  $.ajax({  
    url:"/verifyToken",  
    type:"POST",  
    contentType:"application/json",  
    dataType:"json",  
    data: JSON.stringify({  
      token: token}),  
    success: function(data) {  
      if(data.login==true)  
      {
```

```
result=true;
}
else
{
result=false;
}
},
error: function(data ){
console.log("Wrong Token, Not Authenticated.");
}
});
return result;
}
</script>
<body>
<center>
<h2>Sign In</h2>
<form id="loginForm">
<label for="username">Username:</label><br>
<input type="text" id="username" name="username"><br>
<label for="pwd">Password:</label><br>
<input type="password" id="pwd" name="pwd"><br><br>
<input type="submit" value="Submit">
</form>
</center>
</body>
</html>
```

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>YOGO Solutions</title>
</head>
<body>
<center>
<a href="/signup">Click here to SignUp</a>
<br>
<a href="/login">Click here to Login</a>
<br>
</center>
</body>
</html>
```

SignUp.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>SignUp</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
$('#signUpForm').submit(())=>{
event.preventDefault();
```

```

var name = $("#username").val();
var password=$("#pwd").val();
var work=$("#work").val();

//generateToken by making a service call to /register with post
$.ajax({
  url:"/register",
  type:"POST",
  contentType:"application/json",
  dataType:"json",
  data: JSON.stringify({
    name: name,
    password: password,
    work:work
  }),
  success: function(data) {
    var tokenData=data;
    if(tokenData.signup==false){
      $("#msg").html('<h6>You are Already registered with us!!!</h6><br><a
id="login-link" href="/login">Login</a>');
    }
    else{
      $("#msg").html('<h6>Successfully registered with us!!!</h6><br><a
id="login-link" href="/login">Login</a>');
      $("#login-link").show();
      //window.location.href="/login";
    }
  },
  error: function(data ){

```

```
console.log("Something went wrong");
}
});
});
});
</script>
</head>
<body>
<center>
<h2>SignUp</h2>
<form id="signUpForm">
<label for="username">Username:</label><br>
<input type="text" id="username" name="username"><br>
<label for="work">Profession:</label><br>
<input type="text" id="work" name="work"><br>
<label for="pwd">Password:</label><br>
<input type="password" id="pwd" name="pwd"><br><br>
<input type="submit" value="Submit">
</form>
<div id="msg"></div>
</center>
</body>
</html>
```

Welcome.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
```



```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome Page</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
var user=localStorage.getItem("username");
var job=localStorage.getItem("job");
$(document).ready(function(){
$("#user").append(user);
$("#job").append(job);
});
</script>
</head>
<body>
<center>
<h2>WELCOME TO YOGO SOLUTIONS</h2>
<hr>
<div style="border-style:solid; border-color: blue; width:400px;
height:200px">
<h4 id="user">USER: </h4><br>
<h4 id="job">PROFESSION: </h4></div>
</center>
</body>
</html>

```

Week8: Write MongoDB model for the user and query to fetch user, validate and register.

Database.js:-

```
const {MongoClient} =require('mongodb');
const url = 'mongodb://127.0.0.1:27017';
const client= new MongoClient(url);
let collection;
async function connectDB(dbname,table)
{
let result=await client.connect();
let db= result.db(dbname);
collection = db.collection(table);
console.log("DataBase Connected...");
return collection;
}
exports.getData=async function (name,password)
{
collection = await connectDB("cse","employee");
let response = await
collection.find({name:name,password:password}).toArray();
console.log("From getData method: "+JSON.stringify(response));
collection.close;
return JSON.stringify(response);
}
exports.insertData=async function (emp)
{
collection = await connectDB("cse","employee");
let response=await
collection.insertOne({name:emp.name,work:emp.work,password:emp.pass
word});
console.log("Record inserted Successfully");
collection.close;
```

```
return JSON.stringify(response);  
}
```

Login.html:-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Login Page</title>
```

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

```
<script>
```

```
const form=$("#loginForm");
```

```
$(document).ready(function() {
```

```
$("#loginForm").submit((event)=>{
```

```
event.preventDefault();
```

```
var name = $("#username").val();
```

```
var password=$("#pwd").val();
```

```
//generateToken by making a service call to /auth with post
```

```
$.ajax({
```

```
url:"/auth",
```

```
type:"POST",
```

```
contentType:"application/json",
```

```
dataType:"json",
```

```
data: JSON.stringify({
```

```
name: name,
```

```
password: password
```

```
}),
```

```
success: function(data)
```

```
{
```

```
var tokenData=data;
```

```
if(tokenData.login==true){
  console.log("token data: "+JSON.stringify(tokenData));
  localStorage.setItem("username",tokenData.data[0].name);
  localStorage.setItem("job",tokenData.data[0].work);
  if(verifyLogin(tokenData.token))
  {
    window.location.href="/welcome";
  }
  else{
    alert("Authentication Failed");
  }
}
},
error: function(data ){
  console.log("Something went wrong");
}
});
});
});
function verifyLogin(token)
{
  let result=true;
  $.ajax({
    url:"/verifyToken",
    type:"POST",
    contentType:"application/json",
    dataType:"json",
    data: JSON.stringify({
```

```
token: token,
)),
success: function(data) {
if(data.login==true)
{
result=true;
}
else
{
result=false;
}
},
error: function(data ){
console.log("Wrong Token, Not Authenticated.");
}
});
return result;
}
</script>
<body>
<h2>Sign In</h2>
<form id="loginForm">
<label for="username">Username:</label><br>
<input type="text" id="username" name="username"><br>
<label for="pwd">Password:</label><br>
<input type="password" id="pwd" name="pwd"><br><br>
<input type="submit" value="Submit">
</form>
```

</body>

</html>

Server.js:-

```
const express = require("express");  
const db = require("./database.js");  
const {MongoClient} =require('mongodb');  
var token;  
const jwt = require("jsonwebtoken");  
const app = express();  
const port = 3000;  
app.use(express.json());  
app.get('/',  
(req, res) => {  
res.sendFile(__dirname + '/index.html');  
});  
app.get('/welcome',  
(req, res) => {  
res.sendFile(__dirname + '/welcome.html');  
});  
app.get('/signup',  
(req, res) => {  
res.sendFile(__dirname + '/signup.html');  
});  
app.get('/login',  
(req, res) => {  
res.sendFile(__dirname + '/login.html');  
});  
app.post("/register", async(req, res) => {
```

```
const name = req.body.name;
const password = req.body.password;
const work = req.body.work;
tuple= await db.getData(name,password);
if(JSON.parse(tuple).length>0){
res.json({
  signup: false,
  token: null,
  error: "Already registered",
});
} else {
  const token = jwt.sign(tuple, "secret");
  let emp =
  {
    name: name,
    work: work,
    password: password,
    token:token
  };
  let result=await db.insertData(emp);
  res.json({
    signup: true,
    token: "generated",
    result:result
  });
}
});

app.post("/auth", async (req, res) => {
```

```
const name = req.body.name;
console.log(name);
const password = req.body.password;
console.log(password)
tuple=await db.getData(name,password);
if (JSON.parse(tuple).length>0) {
const token = jwt.sign(tuple, "secret");
res.json({
login: true,
token: token,
data: JSON.parse(tuple),
});
} else {
res.json({
login: false,
error: "please check name and password.",
});
}
});
app.post("/verifyToken", (req, res) => {
const token = req.body.token;
if (token) {
const decode = jwt.verify(token, "secret");
res.json({
login: true,
data: decode,
});
} else {
```



```

res.json({
login: false,
data: "error",
});
}
});
app.post('/welcome',(req, res) => {
res.redirect("/welcome")
});
app.post('/login',(req, res) => {
res.redirect("/login")
});
app.post('/signup',(req, res) => {
res.redirect("/signup")
});
app.listen(port, () => {
console.log(` Server is running :
http://localhost:${port}/`);
});

```

Week9: Design a Single Page Application with different menu items using Angular.

Index.html:-

Add bootstrap CDN links in the index.html

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.c
ss" rel="stylesheet"

integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNllyT2bRjXh0JMHjY6hW+A
LEwIH"

crossorigin="anonymous">

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.
min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jI
eHz"
crossorigin="anonymous"></script>

```

Step 2:-Add RouterModule to the header.component.ts imports

```

import { RouterModule } from '@angular/router';

imports: [RouterModule],

```

Step 3:-Add RouterModule, HeaderComponent to the app.component.ts imports

```

import { RouterModule, RouterOutlet } from '@angular/router';
import { HeaderComponent } from './header/header.component';

imports: [RouterOutlet,RouterModule,HeaderComponent],

```

header.component.html:-

```

<nav class="navbar navbar-expand-lg bg-body-tertiary">
<div class="container-fluid">
<a class="navbar-brand" [routerLink]="['/']">SPA-Demo</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link active" [routerLinkActiveOptions]="{exact:true}"
routerLinkActive="active"
aria-current="page" [routerLink]="['/']">Home</a>

```

```

</li>

<li class="nav-item">

  <a class="nav-link" [routerLinkActiveOptions]="{exact:true}"
  routerLinkActive="active"

  [routerLink]="['/profile']">Profile</a>

</li>

</ul>

</div>

</div>

</nav>

```

home.component.html

```

<section class="container mt-5 text-center">

<h1>Welcome to SPA Demo Page</h1>

</section>

```

profile.component.html

```

<section class="container mt-5 text-center">

<h1>Welcome to Profile Page</h1>

</section>

```

app.component.html

```

<app-header></app-header>

<router-outlet></router-outlet>

```

app.routes.ts

```

import { Routes } from '@angular/router';

import { HomeComponent } from './home/home.component';

import { ProfileComponent } from './profile/profile.component';

export const routes: Routes = [

{

  path:"

```

```
,
  component: HomeComponent,
  title: "Home"
},
{
  path: 'profile',
  component: ProfileComponent,
  title: "Profile"
}
];
```

Week10: Fetch user details from server using REST API and show in profile menu using Angular

Step 1:-

Index.html:-

Add bootstrap CDN links in the index.html

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+A
LEwIH"
crossorigin="anonymous">

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3IHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdSIK1eN7N6jI
eHz"
crossorigin="anonymous"></script>
```

Step 2:-Add RouterModule to the header.component.ts imports

```
import { RouterModule } from '@angular/router';

imports: [RouterModule],
```

Step 3:-Add RouterModule, HeaderComponent to the app.component.ts imports

```
import { RouterModule, RouterOutlet } from '@angular/router';  
import { HeaderComponent } from '../header/header.component';  
imports: [RouterOutlet,RouterModule,HeaderComponent],
```

Step 4:- Create navbar in header.component.html

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">  
  <div class="container-fluid">  
    <a class="navbar-brand" [routerLink]="['/']">SPA-Demo</a>  
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"  
      data-bs-target="#navbarNav" aria-  
      controls="navbarNav" aria-expanded="false" aria-label="Toggle  
      navigation">  
      <span class="navbar-toggler-icon"></span>  
    </button>  
    <div class="collapse navbar-collapse" id="navbarNav">  
      <ul class="navbar-nav">  
        <li class="nav-item">  
          <a class="nav-link active" aria-current="page"  
            [routerLinkActiveOptions]="{exact:true}"  
            routerLinkActive="active" [routerLink]="['/']">Home</a>  
        </li>  
        <li class="nav-item">  
          <a class="nav-link" [routerLinkActiveOptions]="{exact:true}"  
            routerLinkActive="active"  
            [routerLink]="['/profile']">Profile</a>  
        </li>  
      </ul>  
    </div>  
  </div>
```

</nav>

Step 5:-Configure route links in app.routes.ts

```
import { Routes } from '@angular/router';  
import { HomeComponent } from './home/home.component';  
import { ProfileComponent } from './profile/profile.component';  
export const routes: Routes = [  
{  
  path:"  
,  
  component:HomeComponent,  
  title:'Home'  
},  
{  
  path:'profile',  
  component:ProfileComponent,  
  title:'Profile'  
}  
];
```

Step 6:-Use header selector in the app.component.html along with <router-outlet>

<app-header></app-header>

<router-outlet></router-outlet>

Profile.component.html

<div class="card shadow-sm w-50 mt-5 p-2 mx-auto " *ngIf="user">

<h2 class="text-center">Hi, {{ user.name }}</h2>

<table class="table table-bordered mt-3">

```
<tr>
<th>Username</th>
<td>{{user.username}}</td>
</tr>
<tr>
<th>Email</th>
<td>{{user.email}}</td>
</tr>
<tr>
<th>Phone</th>
<td>{{ user.phone }}</td>
</tr>
<tr>
<th>City</th>
<td>{{ user.address?.city }}</td>
</tr>
<tr>
<th>ZIP Code</th>
<td>{{ user.address?.zipcode }}</td>
</tr>
<tr>
<th>Website</th>
<td>{{ user.website }}</td>
</tr>
</table>
</div>
```

Profile.component.ts

```
import { CommonModule } from '@angular/common';
```

```

import { HttpClient, HttpClientModule } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-profile',
  standalone: true,
  imports: [HttpClientModule,CommonModule],
  templateUrl: './profile.component.html',
  styleUrls: ['./profile.component.css']
})
export class ProfileComponent implements OnInit {
  user:any;
  constructor(private http: HttpClient) {}
  ngOnInit(): void {
    this.http.get('https://jsonplaceholder.typicode.com/users/1').subscribe((data: any) => {
      console.log(data);
      this.user = data;
      console.log("API Data::::::::::",data);
    });
  }
}

```

Week11: Design Single Page Application with different menu items using react. □ Layout.js

1. import './Layout.css';
2. import { Outlet, Link } from "react-router-dom";
- 3.
4. const Layout = () => {
5. return (

6. <>

7.

8. <nav className="App-header">

9.

10. <li className="logo">

11. <h1>MyWorld</h1>

12.

13.

14. <Link to="/" className="no">Home</Link>

15.

16.

17. <Link to="/blogs" className="no">Blogs</Link>

18.

19.

20. <Link to="/contact" className="no">Contact</Link>

21.

22.

23. </nav>

24.

25. <Outlet />

26.

27. </>

28.)

29. };

30.

31. export default Layout;

□ Layout.css

1. .App-header{background-color:aqua;}

```
2. li{
3. list-style: none;
4. }
5. ul {
6. display: flex;
7. justify-content: flex-start;
8. gap: 20px;
9. height:50px;
10. }
11.
12. .logo{
13. line-height:10px ;
14.
15. }
16.
17. .no{
18. text-decoration: none;
19. line-height:50px ;
20. font-weight: bold;
21.
22. }
23.
```

□ **Home.js**

```
1. const Home = () => {
2. return <h1>Home</h1>;
3. };
4.
5. export default Home;
```

□ **Contact.js**

1. **const Contact = () => {**
2. **return <h1>Contact Me</h1>;**
3. **};**
- 4.
5. **export default Contact;**

□ **Blogs.js**

1. **const Blogs = () => {**
2. **return <h1>Blog Articles</h1>;**
3. **};**
- 4.
5. **export default Blogs;**

□ **NoPage.js**

1. **const NoPage = () => {**
2. **return <h1>404</h1>;**
3. **};**
- 4.
5. **export default NoPage;**

Open src folder write code inside index.js

□ **index.js**

1. **import React from 'react';**
2. **import ReactDOM from 'react-dom/client';**
3. **import './index.css';**
- 4.
5. **import reportWebVitals from './reportWebVitals';**
6. **import { BrowserRouter, Routes, Route } from "react-router-dom";**
7. **import Layout from "./pages/Layout";**
8. **import Home from "./pages/Home";**

```
9. import Blogs from "../pages/Blogs";
10. import Contact from "../pages/Contact";
11. import NoPage from "../pages/NoPage";
12.
13. const root = ReactDOM.createRoot(document.getElementById('root'));
14. export default function App1() {
15. return (
16. <BrowserRouter>
17. <Routes>
18. <Route path="/" element={<Layout />} />
19. <Route index element={<Home />} />
20. <Route path="blogs" element={<Blogs />} />
21. <Route path="contact" element={<Contact />} />
22. <Route path="*" element={<NoPage />} />
23. </Route>
24. </Routes>
25. </BrowserRouter>
26. );
27. }
28.
29. root.render(
30. <React.StrictMode>
31. <App1 />
32.
33. </React.StrictMode>
34. );
35.
36. reportWebVitals();
```

Week-13 Design web application with different menu items using Django

home.html

1. <!DOCTYPE html>

2. <html lang="en">

3. <head>

4. <meta charset="UTF-8">

5. <meta name="viewport" content="width=device-width, initial-scale=1.0">

6. <title>Home</title>

7. <link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"

rel="stylesheet" integrity="sha384-

QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+ALEwIH"

crossorigin="anonymous">

8. <script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

integrity="sha384-

YvpcrYf0tY3IHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jeHz"

crossorigin="anonymous"></script>

9. </head>

10. <body>

11. <h1 class="text-center">Welcome to CMRIT</h1>

12. <nav class="navbar navbar-expand-lg bg-body-tertiary">

13. <div class="container-fluid">

14. CMRIT

15. **<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">**

16. ****

17. **</button>**

18. **<div class="collapse navbar-collapse" id="navbarNav">**

19. **<ul class="navbar-nav">**

20. **<li class="nav-item">**

21. **Home**

22. ****

23. **<li class="nav-item">**

24. **Dashboard**

25. ****

26. **<li class="nav-item">**

27. **ContactUs**

28. ****

29. ****

30. **</div>**

31. **</div>**

32. **</nav>**

33.

34. **</body>**

35. **</html>**

□ **contactUs.html**

1. **<!DOCTYPE html>**

2. **<html lang="en">**

3. **<head>**

4. **<meta charset="UTF-8">**
5. **<meta name="viewport" content="width=device-width, initial-scale=1.0">**
6. **<title>ContactUs</title>**
7. **</head>**
8. **<body>**
9. **<h1>CMR Institue of Technology,Medchal, Hyderabad.</h1>**
10. **Back to home**
11. **</body>**
12. **</html>**

□ **dashboard.html**

1. **<!DOCTYPE html>**
2. **<html lang="en">**
3. **<head>**
4. **<meta charset="UTF-8">**
5. **<meta name="viewport" content="width=device-width, initial-scale=1.0">**
6. **<title>Dashboard</title>**
7. **</head>**
8. **<body>**
9. **<h1>Welcome to CMRIT Dashboard</h1>**
10. **Back to home**
11. **</body>**
12. **</html>**

Step 9: open project program13 subfolder Write below code

□ **urls.py**

1. **from django.contrib import admin**
2. **from django.urls import path**
3. **from . import index**

```
4. urlpatterns = [
5. #path('admin/', admin.site.urls),
6. path("",index.home,name='home'),
7. path('dashboard/', index.dashboard,name='dashboard'),
8. path('contactUs/', index.contactUs,name='contactUs'),
9. ]
```

□ **settings.py**

```
1. import os
2. templates =>DIRS =>os.path.join(BASE_DIR,'templates')
```

□ **index.py**

```
1. from django.http import HttpResponseRedirect
2. from django.shortcuts import render
3.
4. def home(request):
5. return render(request,'home.html')
6.
7. def dashboard(request):
8. return render(request,'dashboard.html')
9.
10. def contactUs(request):
11. return render(request,'contactUs.html')
```

Step 10: run the server

□ **py manage.py runserver**

Step 11: open browser, run <http://127.0.0.1:8000/>

Week14: Fetch user details from server using REST API and show in profile menu using Django

□ Views.py

```
1. from django.shortcuts import render
2. from django.http import JsonResponse
3. # Create your views here.
4. from rest_framework.response import Response
5. from rest_framework.decorators import api_view
6. from base.models import User
7. from .serializers import UserSerializer
8. import json
9. def home(request):
10. return render(request,'home.html')
11. def register(request):
12. return render(request,'register.html')
13. def profile(request):
14. return render(request,'profile.html')
15. @api_view(['POST'])
16. def getuser(request):
17. req_email=request.data.get('email')
18. Users=User.objects.filter(email=req_email)
19. serializer = UserSerializer(Users,many=True)
20. return Response(serializer.data)
21. @api_view(['POST'])
22. def adduser(request):
23. serializer = UserSerializer(data=request.data)
24. if serializer.is_valid():
25. serializer.save()
26. return Response(serializer.data)
```

□ Urls.py

1. **from django.urls import path**
2. **from . import views**
3. **urlpatterns = [**
4. **path("",views.home),**
5. **path('register/',views.register,name="register"),**
6. **path('profile/',views.profile,name="profile"),**
7. **path('add/', views.adduser,name="add"),**
8. **path('get/',views.getuser,name="get")**
9. **]**

□ Serializers.py

1. **from rest_framework import serializers**
2. **from base.models import User**
3. **class UserSerializer(serializers.ModelSerializer):**
4. **class Meta:**
5. **model = User**
6. **fields = '_all_'**

Step 11: open sub folder program14

Add the below code

□ Urls.py

1. **from django.contrib import admin**
2. **from django.urls import path,include**
3. **urlpatterns = [**
4. **path('admin/', admin.site.urls),**
5. **path("",include('api.urls'))**
6. **]**

□ Settings.py

1. **import os**
2. **installed apps => 'rest_framework',**

3. 'base',

4. templates =>DIRS =>os.path.join(BASE_DIR,'templates')

Step 12: open sub folder base

Add the below code

□ Models.py

1. from django.db import models

2. # Create your models here.

3. class User(models.Model):

4. userid = models.IntegerField()

5. username = models.CharField(max_length=200)

6. occupation = models.CharField(max_length=200)

7. email = models.CharField(max_length=200)

8. created_date = models.DateTimeField(auto_now_add=True)

Again open inside subfolder migrate

Create one file

□ 000_initial.py

Add the below code

1. from django.db import migrations, models

2. class Migration(migrations.Migration):

3. initial = True

4. dependencies = [

5.]

6. operations = [

7. migrations.CreateModel(

8. name='User',

9. fields=[

a. ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False,

```

verbose_name='ID')),
b. ('userid', models.IntegerField()),
c. ('username', models.CharField(max_length=200)),
d. ('occupation', models.CharField(max_length=200)),
e. ('email', models.CharField(max_length=200)),
f. ('created_date', models.DateTimeField(auto_now_add=True)),
10. ],
11. ),
12. ]

```

Step 13: create and open sub folder templates

Add the below code

□ Home.html

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <meta name="viewport" content="width=device-width, initial-
scale=1.0">
6. <title>Home</title>
7. <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
8. <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.c
ss" rel="stylesheet"
integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+A
LEwIH"
crossorigin="anonymous">
9. <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.
min.js"

```

integrity="sha384-

**YvpcrYf0tY3IHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jl
eHz"**

crossorigin="anonymous"></script>

10. </head>

11. <body>

12. <!-- main menu bar -->

13. <nav class="navbar navbar-expand-lg bg-body-tertiary">

14. <div class="container-fluid">

15. CMRIT

**16. <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-**

**target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false"**

aria-label="Toggle navigation">

**17. **

18. </button>

19. <div class="collapse navbar-collapse" id="navbarSupportedContent">

20. <ul class="navbar-nav me-auto mb-2 mb-lg-0">

21. <li class="nav-item">

a. Home

**22. **

23. <li class="nav-item">

a. REGISTER

**24. **

25. <li class="nav-item">

a. PROFILE

**26. **

**27. **

- 28. </div>
- 29. </div>
- 30. </nav>
- 31. </body>
- 32. </html>

□ Register.html

- 1. <!DOCTYPE html>
- 2. <html lang="en">
- 3. <head>
- 4. <meta charset="UTF-8">
- 5. <meta name="viewport" content="width=device-width, initial-scale=1.0">
- 6. <title>Register</title>
- 7. <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
- 8. <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+A LEwIH" crossorigin="anonymous">
- 9. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdSIK1eN7N6jI eHz" crossorigin="anonymous"></script>
- 10. </head>
- 11. <body>
- 12. <!-- registration form -->
- 13. <div id="ucard" class="card shadow-sm w-50 mt-5 p-2 mx-auto ">

```
14. <form id="addUserForm">
15. <div data-mdb-input-init class="form-outline mb-4">
16. <input type="text" id="userid" class="form-control" />
17. <label class="form-label" for="form2Example1">USER ID</label>
18. </div>
19. <div data-mdb-input-init class="form-outline mb-4">
20. <input type="text" id="name" class="form-control" />
21. <label class="form-label" for="form2Example2">USER NAME</label>
22. </div>
23. <div data-mdb-input-init class="form-outline mb-4">
24. <input type="text" id="occ" class="form-control" />
25. <label class="form-label" for="form2Example3">OCCUPATION</label>
26. </div>
27. <div data-mdb-input-init class="form-outline mb-4">
28. <input type="text" id="email" class="form-control" />
29. <label class="form-label" for="form2Example4">EMAIL</label>
30. </div>
31. <button type="submit" data-mdb-button-init data-mdb-ripple-init
class="btn btn-primary btn-block mb-
4">SAVE</button>
32. </form>
33. </div>
34. <div class="w-50 mt-5 p-2 mx-auto ">
35. <h4 id="success" style="display:none">User Added Successfully</h4>
36. <a href="/" data-mdb-button-init data-mdb-ripple-init class="btn btn-
primary btn-block mb-4">BACK</a>
37. </div>
38. </body>
39. <script>
```

```
40. $(document).ready(function() {
41. $('#addUserForm').submit(()=>{
42. event.preventDefault();
43. var userid = $("#userid").val();
44. var name=$("#name").val();
45. var occupation=$("#occ").val();
46. var email=$("#email").val();
47. //save data to django db
48. $.ajax({
49. url:'{% url "add" %}',
50. type:"POST",
51. contentType:"application/json",
52. dataType:"json",
53. data: JSON.stringify({
54. userid:userid,
55. username: name,
56. occupation: occupation,
57. email:email
58. })),
59. success: function(data) {
a. var responseData=data;
b. console.log(responseData)
c. if(responseData!="")
d. {
e. $("#success").show()
f. }
g. },
h. error: function(data ){
```


i. console.log(data);

j. }

k. });

60. });

61. });

62. </script>

63. </html>

□ Profile.html

1. <!DOCTYPE html>

2. <html lang="en">

3. <head>

4. <meta charset="UTF-8">

5. <meta name="viewport" content="width=device-width, initial-scale=1.0">

6. <title>Profile</title>

7. <link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-

QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"

crossorigin="anonymous">

8. <script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-

YvpcrYf0tY3IHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdSIK1eN7N6jeHhZ"

crossorigin="anonymous"></script>

9. <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

10. </head>

11. <body>

12. <!-- Search Bar -->

13. <div class="w-50 mt-5 p-2 mx-auto ">

14. <form id="findUserForm" class="d-flex" role="search">

15. <input id="email" class="form-control me-2" type="search"
placeholder="Search Email" aria-
label="Search">

16. <button class="btn btn-outline-success"
type="submit">Search</button>

17. </form>

18. </div>

19. <!-- profile card template -->

20. <div id="ucard" style="display:none" class="card shadow-sm w-50 mt-5
p-2 mx-auto ">

21.

22. <h2 id="greet" class="text-center">Hi, </h2>

23. <table class="table table-bordered mt-3">

24. <tr>

a. <th>USER ID</th>

b. <td id="uid"></td>

25. </tr>

26. <tr>

a. <th>USERNAME</th>

b. <td id="uname"></td>

27. </tr>

28. <tr>

a. <th>EMAIL</th>

b. <td id="umail"></td>

```
29. </tr>
30. <tr>
  a. <th>OCCUPATION</th>
  b. <td id="uoccu"></td>
31. </tr>
32. </table>
33. </div>
34. <div class="w-50 mt-5 p-2 mx-auto ">
35. <h4 id="err" style="display:none">User Not Found</h4>
36. <a href="/" data-mdb-button-init data-mdb-ripple-init class="btn btn-
  primary btn-block mb-4">BACK</a>
37. </div>
38. </body>
39. <script>
40. $(document).ready(function() {
41. $('#findUserForm').submit(=>{
42. event.preventDefault();
43. $("#ucard").hide();
44. var email=$("#email").val();
45. //find user on django db
46. $.ajax({
47. url:'{% url "get" %}',
48. type:"POST",
49. contentType:"application/json",
50. dataType:"json",
51. data: JSON.stringify({
52. email:email
53. }),
```

```

54. success: function(data) {
a. var responseData=data;
b. console.log(responseData)
c. if(responseData.length!=0)
d. {
e. $("#ucard").show();
f. $("#greet").text(responseData[0].username);
g. $("#uid").text(responseData[0].userid);
h. $("#uname").text(responseData[0].username);
i. $("#umail").text(responseData[0].email);
j. $("#uoccu").text(responseData[0].occupation);
k. }
55. else{
a. $("#err").show();
b. }
c. },
d. error: function(data ){
e. console.log(data);
f. }
g. });
56. });
57. });
58. </script>
59. </html>

```

Step 14:migrate the changes

- ☐ **py manage.py makemigrations**
- ☐ **py manage.py migrate**

Step 15: run the server

□ **py manage.py runserver**

Step 16: open browser, run <http://127.0.0.1:8000/>

.