**PreRequisites:-**

You have to install Node.js and Mongodb to run the application. For now we haven’t touched the database. We are about to decide that.

Download mongodb community edition software from this link:

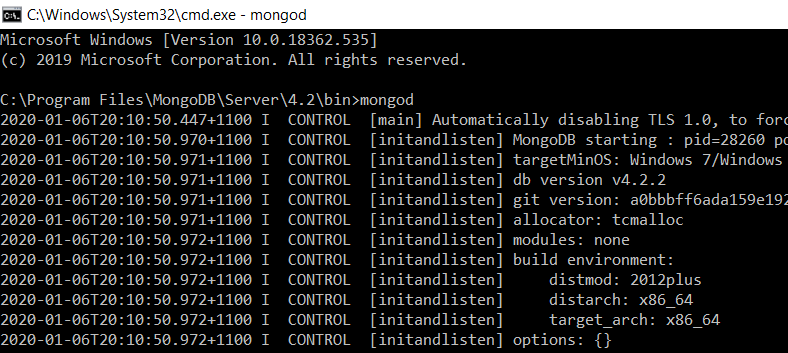
<https://www.mongodb.com/download-center/community>

After installing mongodb navigate to C drive and create a folder named “data”. Then create another folder in data folder as “db”.

NOTE:- We used Local **Mongodb** for our web application. We may change our database from local to online database later.

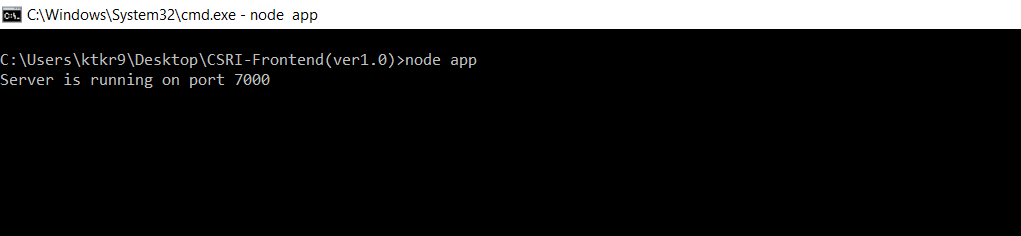
**Steps to run the application:-**

1. Navigate to “C:\Program Files\MongoDB\Server\4.2\bin” in command prompt and start mongodb by typing “mongod”( This is for local mongodb)

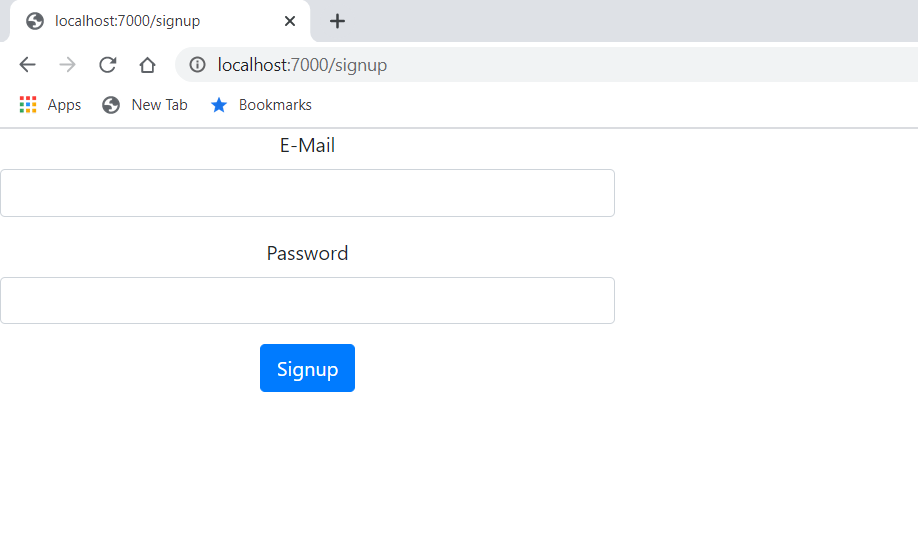


1. Run the app.js file in the command prompt by navigating to the application as **node app**

It shows **Server is running on port 7000**

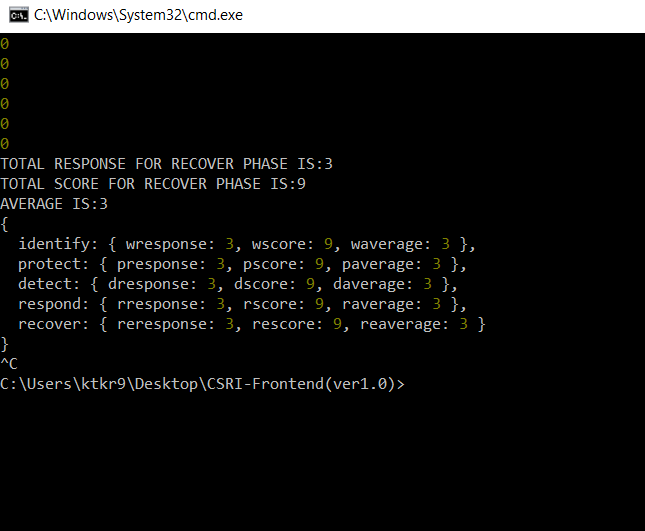


1. Run the application in the url of the browser( like chrome browser, fire fox, Internet Explorer etc) by typing as **localhost:7000/signup**



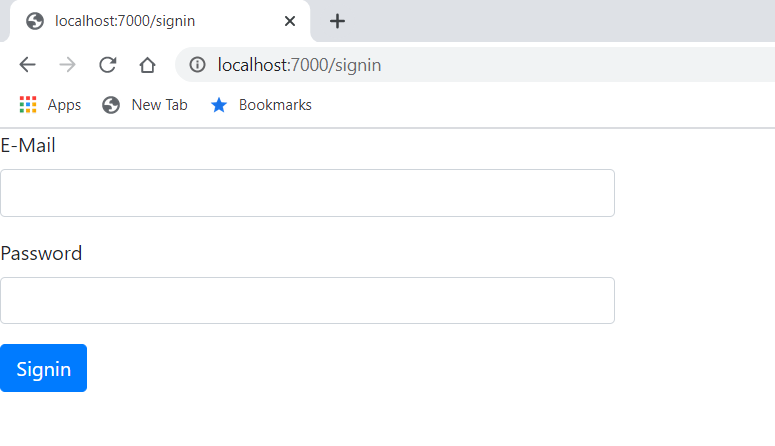
Provide email and password and click on signup. Then you will be navigated to identify phase.

1. Select the option for each question
2. Click next to go to the next phase
3. Continue the process for all the phases
4. Finally you will see the result after answering all the questions in each phase.
5. Close the mongodb command prompt and terminate the node application by pressing **ctrl+c** to stop the server in the command prompt.



**Optional:-**

If you want to see your results again, run the application again and navigate to **localhost:7000/signin**



Enter your details and click on Signin then you will be able to see your results.

**What have we used?**

In this application we have implemented the backend(**app.js**) and other supportive files.

We used **Node.js** as the server.

We used mongodb as the database.

**How do we implemented this?**

For each **question** in each phase, we have **select** tag with **options**.

We used **name** and **id** attributes for every **select** tag. For every **option** tag we have taken **value** as attribute.

**Example:-**

<label for="formGroupExampleInput">1. Disaster Recovery and business continuity plans (DRP) has been created.</label>

<label for="formGroupExampleInput"><i>Annual review of Disaster Recovery and business continuity plans </i></label>

<select class="form-control" id="re1" name="re1">

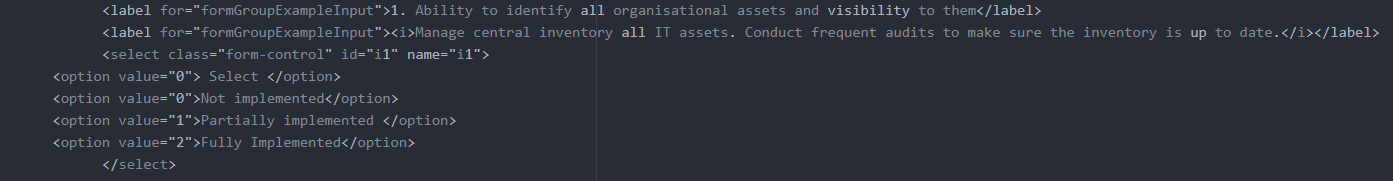
<option value="0"> Select </option>

<option value="1">Not implemented</option>

<option value="2">Partially implemented </option>

<option value="3">Fully Implemented</option>

</select>



**How have we collected data?**

By using the id of each select tag and value of the option selected in the front end we are collecting the data.

This process of collection of data occurs when the user presses the button(i.e., Next for identify, protect, detect, and respond and Submit for recovery phase) at the bottom of the page.

We have implemented a script at the bottom of the html page of each phase( you can see the script by opening the html page of each phase using the notepad or notepad++ or any other editor).

In the script we implemented a function. In the function we used post as the method to take the data to backend(app.js). Post is a secure way to transfer the data from front end to backend.

**Example:-**

<script>

$("#Next").click(function() {

var formData = {

'i1': $('#i1').val(),

'i2': $('#i2').val(),

'i3': $('#i3').val(),

'i4': $('#i4').val(),

'i5': $('#i5').val(),

'i6': $('#i6').val(),

'i7': $('#i7').val(),

'i8': $('#i8').val(),

'i9': $('#i9').val(),

'i10': $('#i10').val(),

'i11': $('#i11').val(),

'i12': $('#i12').val(),

'i13': $('#i13').val(),

'i14': $('#i14').val(),

'i15': $('#i15').val(),

'i16': $('#i16').val(),

'i17': $('#i17').val(),

'i18': $('#i18').val(),

'i19': $('#i19').val(),

'i20': $('#i20').val(),

'i21': $('#i21').val(),

'i22': $('#i22').val(),

'i23': $('#i23').val(),

'i24': $('#i24').val(),

'i25': $('#i25').val()

};

console.log(formData);

$.post("http://localhost:7000/api/identify", formData, function(response) {

console.log("hai")

if (response.error == undefined) {

console.log(response.response);

} else {

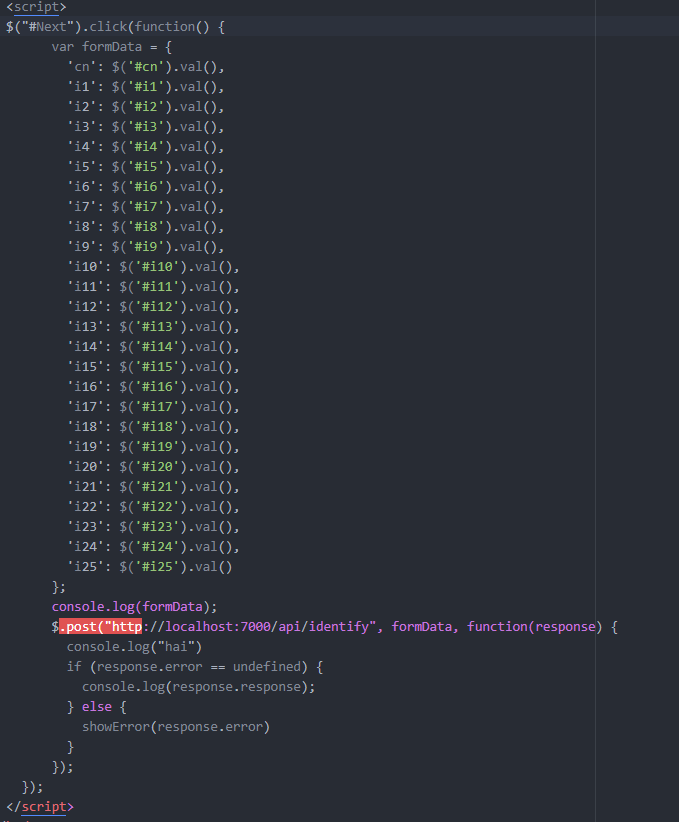
showError(response.error)

}

});

});

</script>



We collect the data using the post method implemented in the app.js file(you can see the code in the app.js file using notepad or notepad++ or any other editor).

**Example:-**

app.post('/api/identify', function(req, res) {

console.log("IDENTIFY PHASE");

var i1=parseInt(req.body.i1);

console.log(i1);

var i2=parseInt(req.body.i2);

console.log(i2);

var i3=parseInt(req.body.i3);

console.log(i3);

var i4=parseInt(req.body.i4);

console.log(i4);

var i5=parseInt(req.body.i5);

console.log(i5);

var i6=parseInt(req.body.i6);

console.log(i6);

var i7=parseInt(req.body.i7);

console.log(i7);

var i8=parseInt(req.body.i8);

console.log(i8);

var i9=parseInt(req.body.i9);

console.log(i9);

var i10=parseInt(req.body.i10);

console.log(i10);

var i11=parseInt(req.body.i11);

console.log(i11);

var i12=parseInt(req.body.i12);

console.log(i12);

var i13=parseInt(req.body.i13);

console.log(i13);

var i14=parseInt(req.body.i14);

console.log(i14);

var i15=parseInt(req.body.i15);

console.log(i15);

var i16=parseInt(req.body.i16);

console.log(i16);

var i17=parseInt(req.body.i17);

console.log(i17);

var i18=parseInt(req.body.i18);

console.log(i18);

var i19=parseInt(req.body.i19);

console.log(i19);

var i20=parseInt(req.body.i20);

console.log(i20);

var i21=parseInt(req.body.i21);

console.log(i21);

var i22=parseInt(req.body.i22);

console.log(i22);

var i23=parseInt(req.body.i23);

console.log(i23);

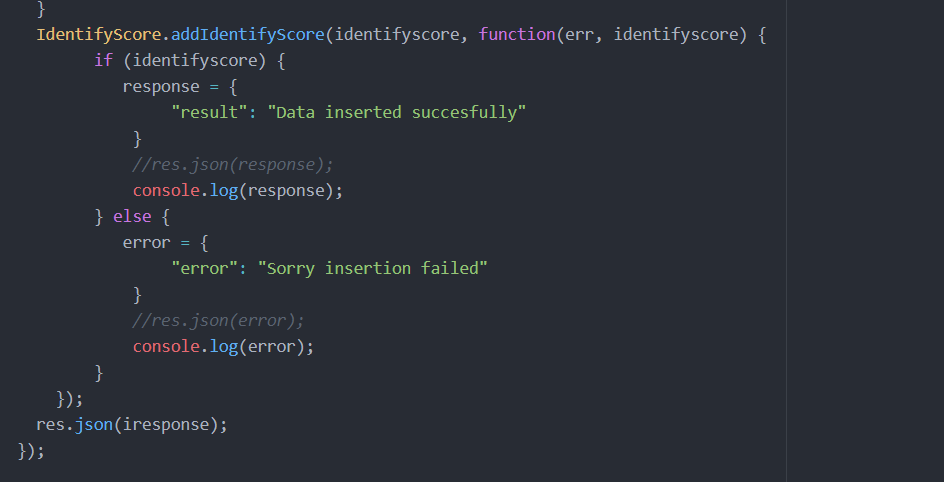
var i24=parseInt(req.body.i24);

console.log(i24);

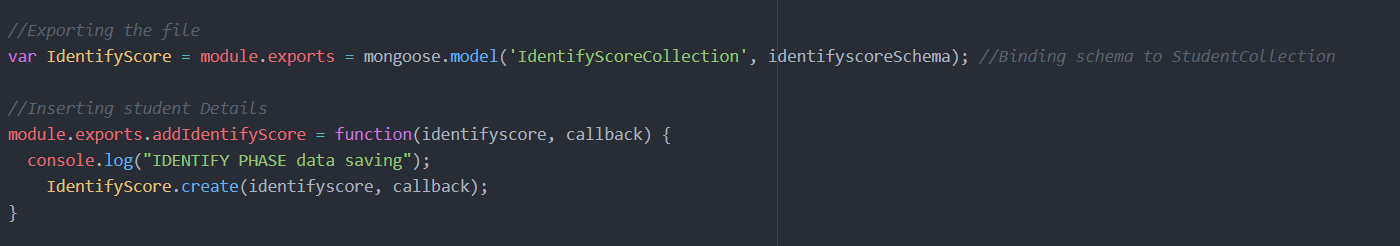
var i25=parseInt(req.body.i25);

console.log(i25);





Then we have calculated the total weights and clientscore for each phase. Then we have stored the data in database and sent the data to the front end through an object by collecting the required data which is calculated for all the phases.



Finally we have displayed the results in the frontend at the result.html page.

