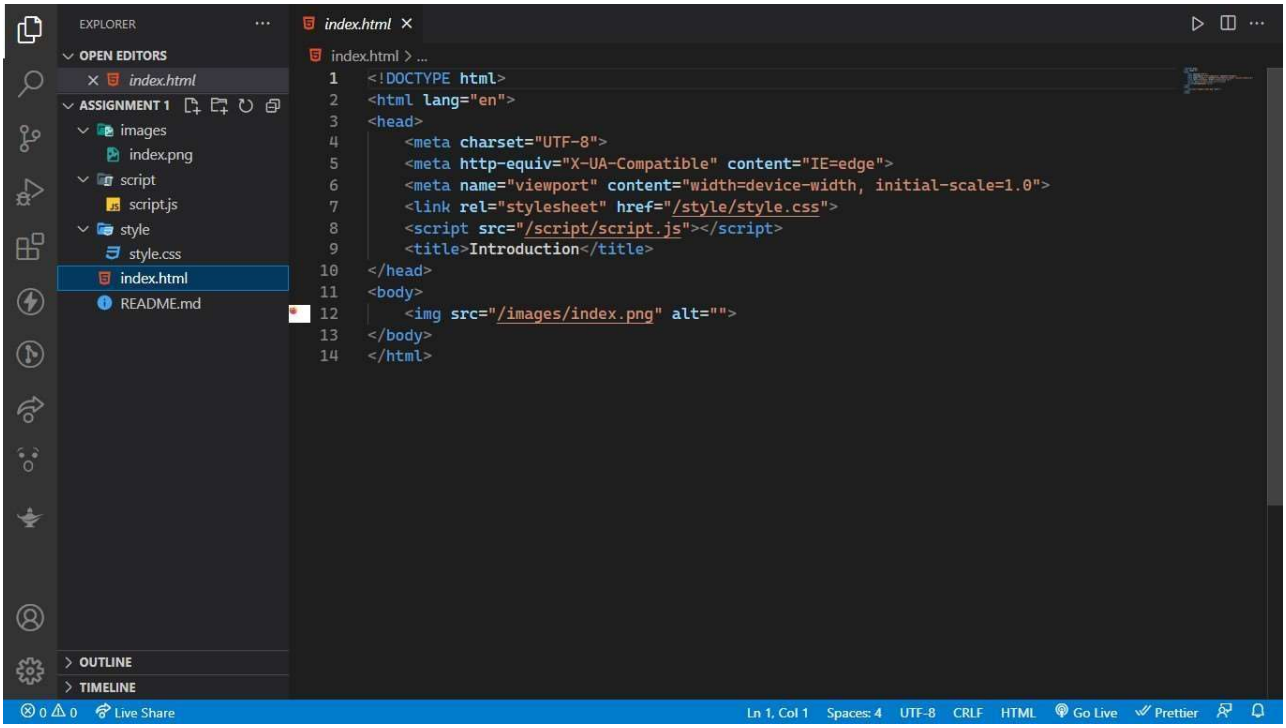
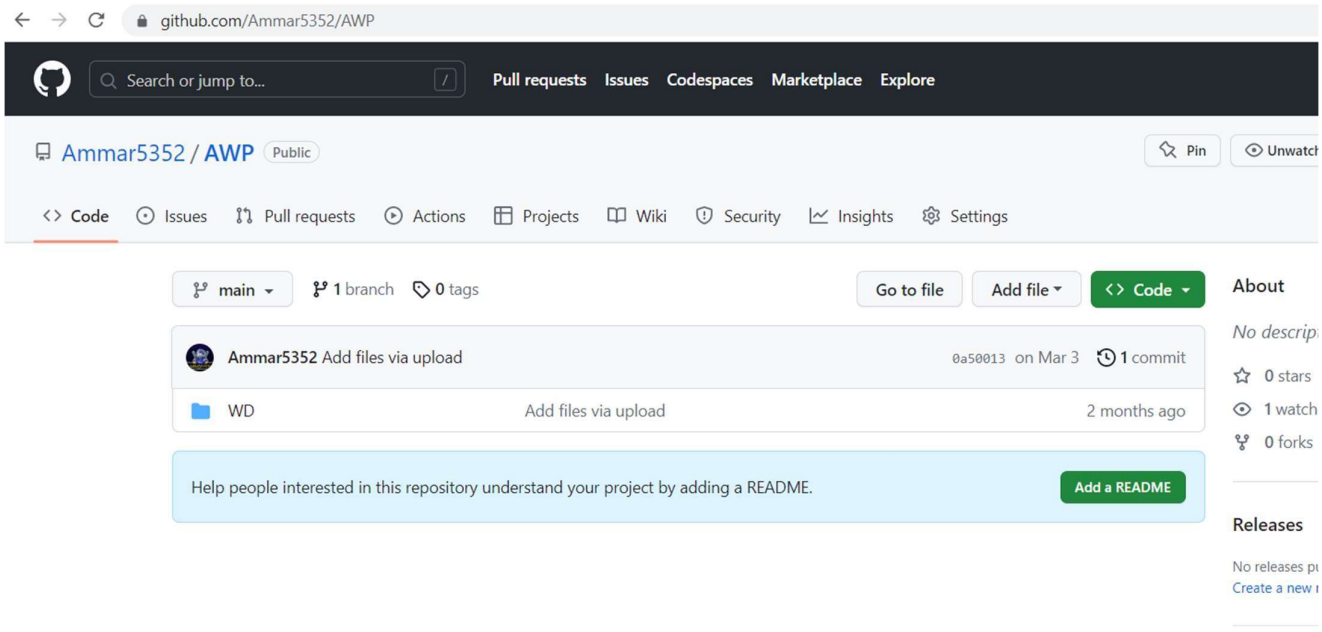
Assignment 1

Perform the Following Steps :

1. Install Visual Studio Code
2. Install git
3. Setup username and email in git
4. Signup in github.com
5. Create a blank repository in github.com
6. Clone that newly created repository in your computer via Visual Studio Code
7. Create directory structure like this :
8. Modify the content index.html so that it looks like this :
9. Push Your Repository to github
10. Submit the Github Repository URL




github.com/Ammar5352/AWP
Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
Ammar5352 / AWP
Public
Pin
Unwatch
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
About
No descrip
Ammar5352 Add files via upload
0a50013 on Mar 3
1 commit
0 stars
1 watch
0 forks
WD
Add files via upload
2 months ago
Help people interested in this repository understand your project by adding a README.
Add a README
Releases
No releases pu
Create a new r

Assignment 2

Create a simple guess the number type game. It should choose a random number between 1 and 100, then challenge the player to guess the number in 10 turns. After each turn the player should be told if they are right or wrong, and if they are wrong, whether the guess was too low or too high. It should also tell the player what numbers they previously guessed. The game will end once the player guesses correctly, or once they run out of turns. When the game ends, the player should be given an option to start playing again.

Submit a Single HTML File.

HTML Code:

```
<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <title>Number Guessing Game</title>

        <style>

            .lastResult {

                color: white;

                padding: 3px;

            }


            body {

                height: 100vh;

                background: rgb(162, 221, 255);

            }


            button {

                width: 160px;
```

```
    padding: 15px 0;
    border-radius: 5px;
    background-color: #000000;
    color: #fff;
    border: none;
    font-size: 18px;
    font-weight: 600;
    margin-bottom: 10px;
}

.form {
    position: absolute;
    width: 50%;
    min-width: 580px;
    transform: translate(-50%, -50%);
    top: 50%;
    left: 50%;
    background: #fff;
    padding: 50px 10px;
    border-radius: 5px;
    display: grid;
    justify-items: center;
    font-family: "Vertigo", sans-serif;
}

input[type="text"] {
    width: 90px;
```

```
            font-weight: 600;
            padding: 10px 0;
            font-size: 28px;
            textalign: center;
            margin-top: 10px;
            margin-bottom: 10px;
            border-radius: 5px;
            border: 2px solid #202020;
            color: #000000;
        }
        .p1 {
            text-align: center;
        }
    </style>
  </head>


 <body>


  <div class="form">
 <h1>Number guessing game</h1>
  <p class="p1">We have selected a random number between 1 and 100. See if you can guess it in 10 turns or fewer. We'll tell you if your guess was too high or too low.</p>
    <label for="guessField">Enter a guess: </label><input type="text" id="guessField" class="guessField">
    <button  onclick="play()" id="my_btn" type="submit" value="Submit Guess" class="guessSubmit">Submit Guess</button>
  <p class="guesses"></p>
    <p class="lastResult"></p>
```

```
    <p class="lowOrHi"></p>
</div>


  <script>       let randomNumber = Math.floor(Math.random()
*     100)     +     1;                              const     guesses     =
document.querySelector('.guesses');              const  lastResult  =
document.querySelector('.lastResult');             const  lowOrHi  =
document.querySelector('.lowOrHi');         const guessSubmit =
document.querySelector('.guessSubmit');       const guessField =
document.querySelector('.guessField');         let guessCount = 1;
let resetButton;      function checkGuess() {        let userGuess =
Number(guessField.value);        if (guessCount
=== 1) {           guesses.textContent = 'Previous
guesses: ';
       }
       guesses.textContent += userGuess + ' ';        if (userGuess
=== randomNumber) {          lastResult.textContent =
'Congratulations! You got it right!';
lastResult.style.backgroundColor = 'green';          lowOrHi.textContent
= '';          setGameOver();
       } else if (guessCount === 10) {
lastResult.textContent = 'GAME OVER!!!';
lowOrHi.textContent = '';       setGameOver();
} else {           lastResult.textContent = 'Wrong!';
lastResult.style.backgroundColor = 'red';
if(userGuess < randomNumber) {
lowOrHi.textContent = 'Last guess was too low!' ;          }
```

```
else if(userGuess > randomNumber) {
lowOrHi.textContent = 'Last guess was too high!';
        }
      }
      guessCount++;        guessField.value
= '';        guessField.focus();
    }
    guessSubmit.addEventListener('click',  checkGuess);
function  setGameOver()  {            guessField.disabled =
true;        guessSubmit.disabled = true;        resetButton
=                              document.createElement('button');
resetButton.textContent      =      'Start      new      game';
document.body.appendChild(resetButton);
resetButton.addEventListener('click', resetGame);
    }
    function resetGame() {       guessCount = 1;       const resetParas
= document.querySelectorAll('.resetParas p');      for(let i = 0 ; i <
resetParas.length ; i++) {           resetParas[i].textContent = '';
      }
      resetButton.parentNode.removeChild(resetButton);
guessField.disabled = false;        guessSubmit.disabled = false;
guessField.value = '';        guessField.focus();
lastResult.style.backgroundColor = 'white';
randomNumber = Math.floor(Math.random() * 100) + 1;
    }
  </script>
 </body>  </html>
```

Assignment 3

Perform Following Steps :

1. Create an HTML file (e.g. first_page.html) that specifies a page that contains a heading and two paragraphs of text. Use the HTML tags <h1>,</h1>, , and in this exercise. As the texts in the heading and paragraphs you can use any texts you like. 2. Add an unordered list to your first web page. An unordered list should look like the following when it is shown by a browser: An unordered list can be specified with the tags <ul> and . An unordered list typically contains a number of list items that can be specified with tags <li> and . After you have created your unordered list, check out what happens when you convert it to an ordered list by replacing the tags <ul> and with <ol> and , respectively.

3. Add an image to your web page. In this exercise you must use the <img> tag. As an image, you can use any .jpg or .png file you find on the Internet.

4. Create another .html file that contains a heading and a couple of paragraphs. You could name this new file another_page.html, and you should place it into the same folder where your first .html is. After you have created the new .html page, add a link to the first page so that the browser will load another_page.html when you click the text Go to the other page. in the first page.

5. HTML tags like <a> can have certain attributes. The href attribute is mandatory in the <a> tag. Additionally it is possible to use the title attribute which specifies a text that emerges when the mouse cursor is moved above a link. This kind of text is called a tool tip. Modify the link that you created in the previous exercise so that a tool tip says "This leads you to another page." when the mouse cursor is over the link.

6. It is possible to use a picture (image) as a link. Modify your page so that the picture that is on your page will also serve as a link that leads to another page.

7. Upload Two .html files on moodle.

HTML Code:

first_page.html

```
<!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>Assignment 3</title>
    <style>
      /* Tooltip container */
      .tooltip {
        position: relative;
        display: inline-block;
        borderbottom: 1px dotted black;
        /* If you want dots under the hoverable text */
      }
      /* Tooltip text */
      .tooltip .tooltiptext {
        visibility: hidden;
        width: 120px;
        background-color: #555;
        color: #fff;
        text-align: center;
        padding: 5px 0;
        border-radius: 6px;

        /* Position the tooltip text */
        position: absolute;
        z-index: 1;
        bottom: 125%;
        left: 50%;
        margin-left: -60px;

        /* Fade in tooltip */
        opacity: 0;
        transition: opacity 0.3s;
      }
      /* Tooltip arrow */
      .tooltip .tooltiptext::after {
        content: "";
        position: absolute;
        top: 100%;
        left: 50%;
        margin-left: -5px;
```

```
            border-width: 5px;
            borderstyle: solid;
            border-color: #555 transparent transparent transparent;
        }

        /* Show the tooltip text when you mouse over the tooltip container */
        .tooltip:hover .tooltiptext {
            visibility: visible;
            opacity: 1;
        }
    </style>
</head>

<body>
    <!--Task 1-->
    <h1>Student</h1>
    <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Perferendis, laboriosam! Minus nesciunt at possimus quasi! Ducimus
        maxime velit mollitia sed sapiente perspiciatis sunt nihil
        temporibus. Totam voluptatem ex dolores distinctio tenetur a
        inventore harum.
    </p>
    <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Nulla
        beatae, ipsum voluptatem ipsam excepturi architecto deserunt tenetur
        aliquam magni saepe vel facilis, consequuntur nemo explicabo dolorum
        autem quam fuga illo porro facere voluptatibus hic. Sequi cum
        voluptate, cupiditate atque sunt inventore itaque quisquam sint
        possimus necessitatibus delectus, neque molestias error vitae quis
        accusamus laborum.
    </p>
    <hr />
    <!--Task 2-->
    <ul type="square">
        <li>First</li>
        <li>Second</li>
        <li>Third</li>
        <li>Fourth</li>
        <li>Fifth</li>
        <li>Sixth Para</li>
    </ul>
    <ol type="I">
        <li>First</li>
        <li>Second</li>
        <li>Third</li>
```

```
        <li>Fourth</li>
        <li>Fifth</li>
        <li>Sixth Para</li>
    </ol>
    <hr />
    <!--Task 3-->
    <a href="/another_page.html"
      ><img
        height="500px"
        width="1000px"
        src="https://images.unsplash.com/photo-1498050108023-c5249f4df085?ixlib=rb-
4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&f i
t=crop&w=1472&q=80"
        alt="This is an image."
    /></a>
    <!-- <a href="/another_page.html"><img height="500px" width="1000px" src="/i1.jpg"
alt="This is an image."></a> -->

    <!--Task 4 & 5-->
    <hr />
    <div>
      <div class="tooltip">
        <a href="/another_page.html">Cilck Here</a>
        <span class="tooltiptext">This leads you to another page.</span>
      </div>
    </div>
  </body>
</html>
```

Another_page.html

```
<!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>Assignment 3 New</title>
  </head>
  <body>
    <h1>This is a Another Page</h1>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
    <p>
```

```
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Illum,
      consequatur.
    </p>
  </body>
</html>
```

Assignment 4 - Create web application in Nodejs

Create web application in nodejs to display "hello world" in browser

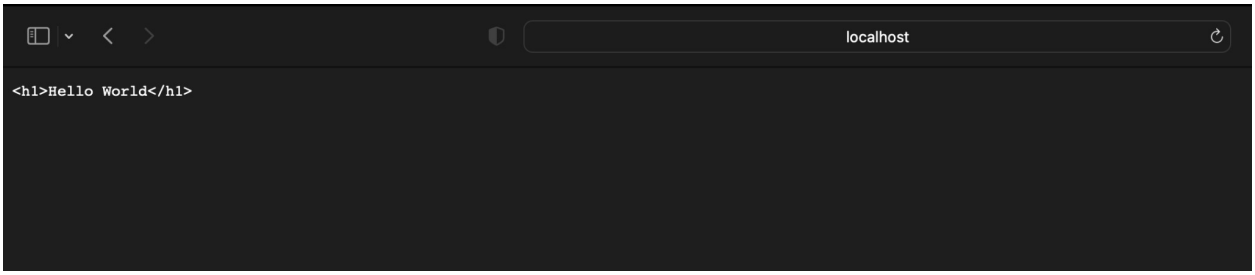
Vs Code:

Hello.js

```
var http = require("http");

http.createServer(function (req, res) { res.end("node js
hello world !!! ");
  }).listen(6100);

console.log("Server is running live at http://127.0.0.1:6100/"); console.log("node js
hello world");
```



Assignment 5

Read file " myfile.txt" in nodejs using synchronous and asynchronous file function and find out How much time saved by non -blocking asynchronous I/O operation.

Note: Create separate nodejs program for blocking and non-blocking I/O Txt file:

This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs
This is NodeJs

Blocking I/0:

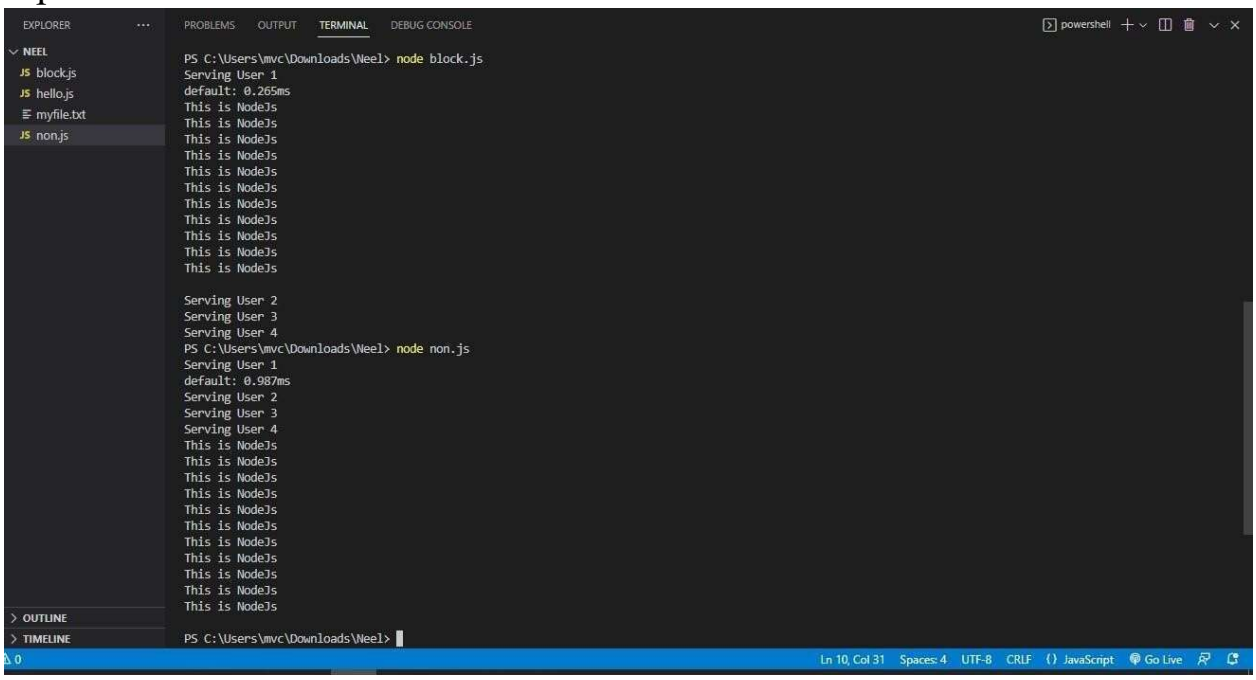
```
var fs = require('fs'); console.log("Serving User 1");
console.time();

var data = fs.readFileSync('myfile.txt');
console.log(data.toString()); console.timeEnd();
console.log("Serving user 2") console.log("Serving user 3");
console.log("Program ended");
```

Non Blocking I/0:

```
console.time();  var fs = require('fs');
console.log("Serving User 1");
console.time();
var data = fs.readFile('myfile.txt', function (err, data) {
        if (err) return console.error(err); console.log(data.toString());
});  console.timeEnd();
console.log("Serving User 2"); console.log("Serving User 3");
console.log("Serving User 4")
```

Output of Both Files :



Assignment – 6

Create and publish your own Node package to display current date and time with appropriate message

Step-1 : Create npm account by going to npmjs.com

Step-2: Go to cmd and do login into npm(node package manager) account by npm login and it will ask username, password, email, OTP authenticator etc.

```
C:\Users>npm login
npm notice Log in on https://registry.npmjs.org/
Login at:
https://www.npmjs.com/login?next=/login/cli/942c0aa5-c18e-4a24-936f-7f6bd29ae39e
Press ENTER to open in the browser...

Logged in on https://registry.npmjs.org/.
```

Step-3 : Create directory for module that you will publish and create a new package using npm init

```
C:\Users\Admin>mkdir Ammartime

C:\Users\Admin>npm init
```

```
C:\Users\Admin\Ammartime>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible default

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (ammartime)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\Admin\Ammartime\package.json:

{
  "name": "ammartime",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
```

```
{
  "name": "ammartime",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes)

C:\Users\Admin\Ammartime>
```

Step-4 : As entry point : index.js , so you have to write your code in index.js

```js
function date() {
  date = new Date();
  var month = date.getMonth();
  month++;
  console.log(date.getDate() + "-" + month + "-" + date.getFullYear());
  if (date.getHours() <= 12) {
    if (date.getHours() == 12) {
      console.log(`${date.getHours()}:${date.getMinutes()} PM `);
    } else console.log(`${date.getHours()}:${date.getMinutes()} AM `);
  }
  if (date.getHours() > 12) {
    console.log(`${date.getHours() % 12}:${date.getMinutes()} PM `);
  }
}

date();
```

Output :

```
PS C:\Users\Admin\Ammar\Ammartime> node index.js
11-3-2023
10:32 AM
PS C:\Users\Admin\Ammar\Ammartime>
```

Output of package:

-

```
PS C:\Users\Admin\Ammar\Ammartime> node check-package.js
11-3-2023
10:40 AM
This is for package validation
PS C:\Users\Admin\Ammar\Ammartime>
```

Assignment 7 - AngularJS Part 1

Create a Hello World Web Page using Angular JS.

The Web page should display the message "Hello world" by using the model value from the controller. Also make sure that the controller should be a part of a Module.

Create a Angular JS Application which displays the output like this :

Name : James Bond
Age : XX
Mobile : 85475995858
Emails
Description : official Email : james.bond@bond.com
Description : personal1 Email : bond******@yahoo.com
Description : personal2 Email : bond******@gmail.com

HTML Code:

Index.html

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

<head>
    <meta charset="utf 8">
    <title>Hello World</title>
    <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script type="text/javascript" src="app.js"></script>
</head>

<body ng-app="myApp">
```

```
    <style>
body{ background-
color:aliceblue;
        }
    </style>

     <div ng-controller="HelloWorldCtrl">



       <table border="1">
          <tr>
             <td>
                Name : {{person.name}}
             </td>
          </tr>
          <tr>
           <td>
              Age : {{person.age}}
           </td>
        </tr>
        <tr>
           <td>
              Mobile : {{person.mobile}}
           </td>
        </tr>
        <tr>
           <td>
              Email IDs
           </td>
```

```
        </tr>
        <tr ng-repeat = "email in person.emails">
            <td >
                Description : {{email.discription}}  Email : {{email.email}}
            </td>
        </tr>
       </table>
    </div>



</body>


</html>
```

Index-copy.html

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

<head>
    <meta charset="utf 8">
    <title>Hello World</title>
    <script                                                    type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script type="text/javascript" src="app.js"></script>
</head>

<body ng-app="myApp">
```

```
<div ng-controller="HelloWorldCtrl">

    <table border="1">
      <tr>
        <td>
            Name : <input type="text" ng-model="person.name">
        </td>
      </tr>
     <tr>
      <td>
          Age : <input type="number" ng-model="person.age">
      </td>
    </tr>
    <tr>
      <td>
          Mobile : <input type="text" ng-model="person.mobile">
      </td>
    </tr>
    <tr>
<td>
          Emails
      </td>
    </tr>
    <tr ng-repeat = "email in person.emails">
      <td >
          Description : <input type="text" ng-model="email.discription">
          Email : <input type="email" ng-model="email.email">
      </td>
    </tr>
```

```
        </table>
<hr>
        <table border="1">
          <tr>
              <td>
                  Name : {{person.name}}
              </td>
          </tr>
         <tr>
           <td>
              Age : {{person.age}}
           </td>
        </tr>
        <tr>
           <td>
               Mobile : {{person.mobile}}
           </td>
        </tr>
        <tr>
<td>
               Emails
           </td>
        </tr>
        <tr ng-repeat = "email in person.emails">
           <td >
               Description : {{email.description}}
               Email : {{email.email}}
           </td>
        </tr>
```

```
      </table>
    </div>



</body>
</html>
```

Js Code:

app.js

```
var myApp = angular.module("myApp", []); myApp.controller("HelloWorldCtrl", function
($scope)
{
    $scope.person =
     {
        name : "AMMAR TAPIA",
        age : "21",
        mobile : "XXXXX-XXXXX",
        emails : [
            {description:'College ',email:'ammartapia.it20@scet.ac.in'},
            {description:'Personal 1',email:'jbond***@yahoo.com'},
            {description:'Personal 2',email:'james***@gmail.com'}
        ]
    };
});
```

Assignment 8 - AngularJS Part 2

Create a Sign Up Form in AngularJS.
The Sign up form should ask for Name, Mobile, Email, Address, Date of Birth, Gender, Username , Password and Confirm Password.
The Form should have two buttons, one for reset and one for submit. The Submit Button should be enabled only when All the input values are validated.

HTML Code:

Angular_form.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Registration Form</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>     <link
rel="stylesheet" href="css/style.css" />
  </head>
  <body style="margin-left: 100px">
    <h1 class="box">Registration Form</h1>
    <div ng-app="registrationform">
      <form        action="information.html"
method="get"
name="registrationform"
class="container"        novalidate
      >
        <label for="name">Name: </label>
        <input type="text" name="name" id="name" ng-model="name" required />
```

```
<span style="color: red" ng-
show="registrationform.name.$dirty
&& registrationform.name.$invalid"
>
<span ng-show="registrationform.name.$error.required"
>Name is required.</span
>
</span>
<br /><br />
<label for="mobile">Mobile Number: </label>
<input type="number"
name="mobile" id="mobile"
maxlength="10" minlength="10" ng-
model="mobile" required /> <span
style="color: red"
ngshow="registrationform.mobile.$dirty
&& registrationform.mobile.$invalid"
>
<span ngshow="registrationform.mobile.$error.required"
>Mobile Number is required.</span
>
<span ngshow="registrationform.mobile.$error.mobile"
>Invalid Mobile Number.</span
>
</span>
<br /><br />
<label for="email">email: </label>
<input type="email" name="email" id="email" ng-model="email" required /> <span
style="color: red" ngshow="registrationform.email.$dirty
```

```
&& registrationform.email.$invalid"
      >
        <span ngshow="registrationform.email.$error.required"
          >Email is required.</span
        >
        <span ng-show="registrationform.email.$error.email"
          >Invalid email address.</span
        >
      </span>
      <br /><br />
      <label for="address">Address: </label>
      <textarea
rows="5"          cols="30"
id="address"
name="address"
ngmodel="address"
required
      ></textarea>        <span          style="color: red"
ngshow="registrationform.address.$dirty && registrationform.address.$invalid"
      >
        <span ngshow="registrationform.address.$error.required"
         >Address is required.</span
       >
      </span>
      <br /><br />
      <label for="dob">Date of Birth: </label>
      <input type="date" name="dob" id="dob" ng-model="dob" required />
      <span          style="color: red"          ng-
show="registrationform.dob.$dirty && registrationform.dob.$invalid"
```

```
>
  <span ng-show="registrationform.dob.$error.required"
    >Date of Birth is required.</span
  >
</span>
<br /><br />
<label>Gender: </label>
<label for="male">Male</label>
<input type="radio"
name="gender"
id="male"
ngmodel="gender"
value="Male"
/>
<label for="female">Female</label>
<input type="radio"
name="gender"
id="female"
ngmodel="gender"
value="Female" />

<label for="other">Other</label>
<input
type="radio"
name="gender"
  id="other"
ngmodel="gender"
value="Other"
/>
<span style="color: red" ngshow="registrationform.gender.$dirty
```

```
&& registrationform.gender.$invalid"
        >
          <span ng-show="registrationform.gender.$error.required"
            >Gender is required.</span
          >
        </span>
        <br /><br />
        <label for="uname">Username: </label>
        <input type="text" name="uname" id="uname" ng-model="uname" required />
        <span          style="color: red"          ng-
show="registrationform.uname.$dirty && registrationform.uname.$invalid"
        >
          <span ngshow="registrationform.uname.$error.required"
            >Username is required.</span
          >
        </span>
        <br /><br />
        <label for="password">Password: </label>
        <input    type="password"
       name="password"
          id="password"
ngmodel="password"
          required
/>
        <span          style="color: red"          ng-
show="registrationform.password.$dirty
&& registrationform.password.$invalid"
        >
          <span ng-show="registrationform.password.$error.required"
```

```
    >Password is required.</span >
  </span>
  <br /><br />
  <label for="cpassword">Confirm Password: </label>
  <input type="password" name="cpassword"
id="cpassword" ngmodel="cpassword"
equal="{{ registrationform.password }}" required />
  <span style="color: red"
  ngshow="registrationform.cpassword.$dirty
  && registrationform.cpassword.$invalid"

  >
    <span ng-show="registrationform.cpassword.$error.required" >Password
Confirmation is required.</span
    >
    <span ng-show="registrationform.cpassword.$error.equal"
      >Password does nor match above.</span
    >
  </span>
  <br /><br /> <input type="submit"
value="Submit"
onclick="passvalues(this.form);"
ngdisabled="(registrationform.name.$invalid) ||
(registrationform.mobile.$invalid) || (registrationform.email.$invalid) ||
(registrationform.address.$invalid) || (registrationform.dob.$invalid) ||
(registrationform.gender.$invalid) || (registrationform.uname.$invalid) ||
(registrationform.password.$invalid) ||
(registrationform.cpassword.$invalid)"
  />
  <input type="reset" value="Reset" />
```

```
    </form>
  </div>    <script>      var registrationform =
angular.module("registrationform", []); function passvalues() {  var
name = document.getElementById("name").value;
localStorage.setItem("setiteam", name);
      // console.log(name1) var mobile =
      document.getElementById("mobile").value;
      localStorage.setItem("MobileNumber", mobile);        var email
      = document.getElementById("email").value;
      localStorage.setItem("email", email);        var address =
      document.getElementById("address").value;
      localStorage.setItem("ad", address);        var dob =
      document.getElementById("dob").value;
      localStorage.setItem("db", dob);        var male =
      document.getElementById("male").value;
      localStorage.setItem("ma", male);        var uname =
      document.getElementById("uname").value;
      localStorage.setItem("un", uname);        var password =
      document.getElementById("password").value;
      localStorage.setItem("pass", password);        var cpassword =
      document.getElementById("cpassword").value;
      localStorage.setItem("cpass", cpassword);
      return false;
    }
  </script>
 </body>
</html>
```

Angular_form_display.html

```
<!-- <!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>information page</title>
</head>
<body>
My name is :<span id="box"> </span><br> My mobile
number: <span id="box1"></span><br> my email is id:
<span id="box2"></span><br>
Address:<span id="box4"></span><br>
Date of Birth:<span id="box5"></span><br>
Gender:<span id="box6"></span><br> Username:<span
id="box9"></span><br> password<span
id="box10"></span><br> confirm password<span
id="box11"></span><br>
<script>  document.getElementById("box").innerHTML=localStorage.getItem("setiteam")
document.getElementById("box1").innerHTML=localStorage.getItem("MobileNumber")
document.getElementById("box2").innerHTML=localStorage.getItem("email")
document.getElementById("box4").innerHTML=localStorage.getItem("ad")
document.getElementById("box5").innerHTML=localStorage.getItem("db")
document.getElementById("box6").innerHTML=localStorage.getItem("ma")
document.getElementById("box9").innerHTML=localStorage.getItem("un")
document.getElementById("box10").innerHTML=localStorage.getItem("pass")
```

```
document.getElementById("box11").innerHTML=localStorage.getItem("cpass") </script>
</body>
</html> -->
```

Assignment 9: Implement Node js program to handle divide by zero exception

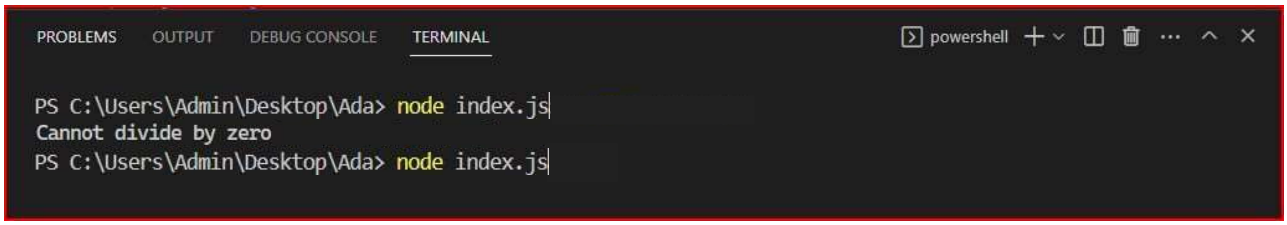
Js Code:

Index.js

```
const divide = (a, b, next) => {     if(b==0) {         next(new
Error("Cannot divide by zero"));
   } else {         next(null,
a/b);
   }
}

divide(5, 0, (error, res) =>{
   if(error) {
console.log(error.message);
   } else {
console.log(res);
   }
});
```

Output:



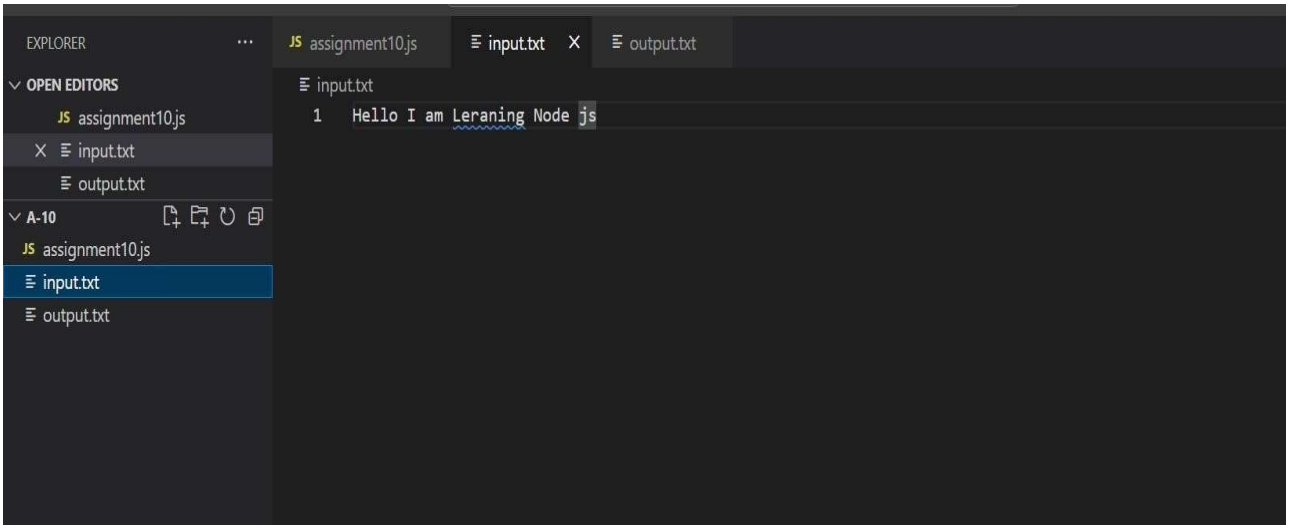
Assignment 10

Implement Node js program to read data from file "input.txt" and write these data to "output.txt" using concept of streams.

Node Js Code :-

```
const fs = require('fs'); const readStream = fs.createReadStream('input.txt', { encoding:
'utf8' }); const writeStream =
fs.createWriteStream('output.txt', { encoding: 'utf8' }); readStream.on('data',
(chunk) => { writeStream.write(chunk);
}); readStream.on('end', () => { writeStream.end();
});
writeStream.on('finish', () => {
console.log('Data has been written to output.txt'); });
```

Input.txt File:



Output of Code :-

```js
const fs = require('fs');
const readStream = fs.createReadStream('input.txt', { encoding: 'utf8' });
const writeStream = fs.createWriteStream('output.txt', { encoding: 'utf8' });
readStream.on('data', (chunk) => {
writeStream.write(chunk);
});
readStream.on('end', () => {
writeStream.end();
});
writeStream.on('finish', () => {
console.log('Data has been written to output.txt');
});
```

```
PS C:\Sem 6 Material\AWP\Practicals\a-10> node "c:\Sem 6 Material\AWP\Practicals\a-10\assignment10.js"
Data has been written to output.txt
PS C:\Sem 6 Material\AWP\Practicals\a-10>
```

Output.txt File:

```
Hello I am Leraning Node js
```

Assignment 11

Create a Single Page Application using AngularJS.

```
<!DOCTYPE html>

<html ng-app="myApp">

<head>

	<script src=
"https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular.min.js">
	</script>
	<script src=
"https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular-route.min.js">
	</script>

	<style>
		body {
			text-align: center;
			font-family: Arial, Helvetica, sans-serif;
		}

		h1 {
			color: green;
		}
	</style>
</head>

<body>
	<h1>Jason</h1>
	<h3>Single Page Application in AngularJS</h3>


	<script type="text/ng-template" id="first.html">
		<h1>First Page</h1>
		<h2 style="color:green">
			Hello This is Jason
		</h2>
		<h3>{{message}}</h3>
	</script>
	<script type="text/ng-template" id="second.html">
```

```
        <h1>Second Page</h1>
        <h2 style="color:green">
    I am an It Engineer
        </h2>
        <h3>{{message}}</h3>
</script>
<script type="text/ng-template" id="third.html">
        <h1>Third Page</h1>
        <h2 style="color:green">
 Contact me
        </h2>
        <h3>{{message}}</h3>
</script>

<a href="#/">First</a>
<a href="#/second">Second</a>
<a href="#/third">Third</a>


<div ng-view></div>

<script>
        var app = angular.module('myApp', []);
        var app = angular.module('myApp', ['ngRoute']);
        app.config(function ($routeProvider) {
                $routeProvider
                        .when('/', {
                                templateUrl: 'first.html',
                                controller: 'FirstController'
                        })

                        .when('/second', {
                                templateUrl: 'second.html',
                                controller: 'SecondController'
                        })

                        .when('/third', {
                                templateUrl: 'third.html',
                                controller: 'ThirdController'
                        })

                        .otherwise({ redirectTo: '/' });
        });
```

```
            app.controller('FirstController', function ($scope) {
                $scope.message = 'Hello from FirstController';
            });
            app.controller('SecondController', function ($scope) {
                $scope.message = 'Hello from SecondController';
            });
            app.controller('ThirdController', function ($scope) {
                $scope.message = 'Hello from ThirdController';
            });
        </script>
</body>

</html>
```