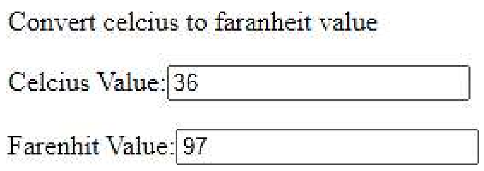
Convert Value Celsius into Fahrenheit in JavaScript

Source Code

```
<!DOCTYPE html>
<html>
    <body>
        <p>Convert celcius to faranheit value</p>
        <p>Celcius Value:<input id="txtCelcius" onkeyup="convert('C')" /></p>
        <p>Farenhit Value:<input id="txtFahrenhit" onkeyup="convert('F')" /></p>
        <script>
            function convert(degree) {var p;
                if (degree == "C") {
                    p = (document.getElementById("txtCelcius").value * 9) / 5 + 32;
                    document.getElementById("txtFahrenhit").value = Math.round(p);
                } else {
                    p = ((document.getElementById("txtFahrenhit").value - 32) * 5) / 9;
                    document.getElementById("txtCelcius").value = Math.round(p);
                }
            }
        </script>
    </body>
</html>
```

Output:



Convert celcius to faranheit value

Celcius Value: 36

Farenhit Value: 97

Results:

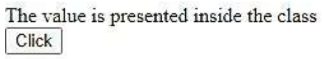
Thus, the above Program to Convert Value Celsius into Fahrenheit in JavaScript was executed successfully.

Implement getElementsByClassName() method in JavaScript

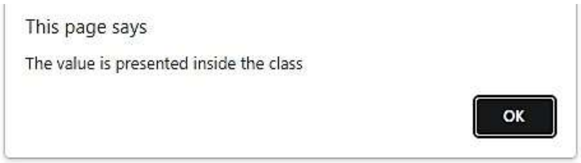
Source Code

```
<!DOCTYPE html>
<html>
  <body>
    <div class="myclass">The value is presented inside the class</div>
    <button onclick="RetriveByClassName()" id="btnClick">Click</button>
    <script>
      function RetriveByClassName() {
        var x = document.getElementsByClassName("myclass");alert(x[0].innerHTML);
      }
    </script>
  </body>

</html>
```

Output:



The value is presented inside the class



This page says

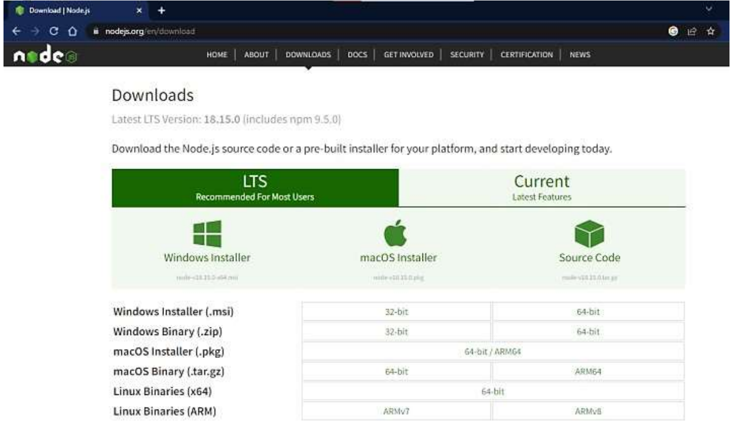
The value is presented inside the class

Results:

Thus, the above Program to Implement getElementsByClassName() method in JavaScript was executed successfully.

Install Node.JS

Step 1:



Download | Node.js
nodejs.org/en/download
HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

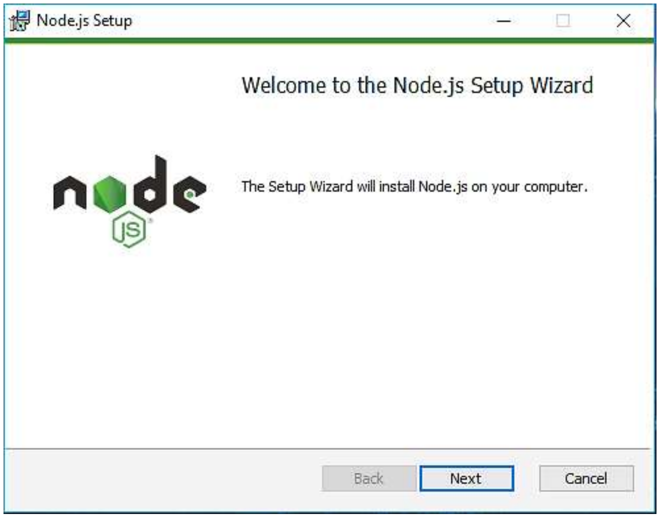
Downloads

Latest LTS Version: 18.15.0 (includes npm 9.5.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8

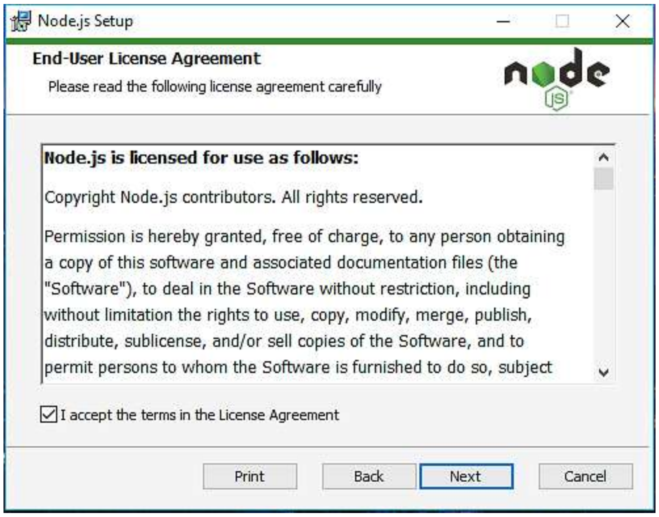
Step 2:



Welcome to the Node.js Setup Wizard

The Setup Wizard will install Node.js on your computer.

Step 3:



End-User License Agreement

Please read the following license agreement carefully

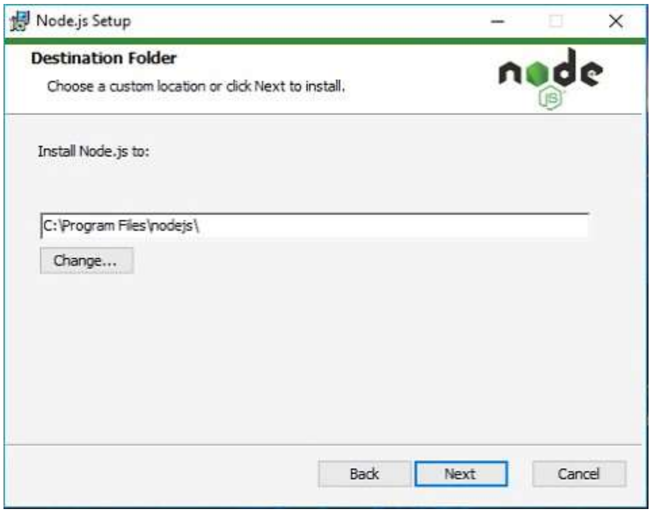
Node.js is licensed for use as follows:

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject

☑ I accept the terms in the License Agreement

Step 4:



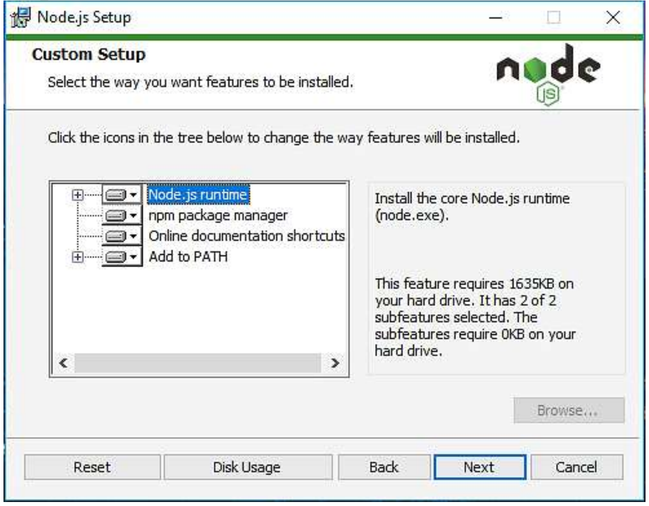
Destination Folder

Choose a custom location or click Next to install.

Install Node.js to:

C:\Program Files\nodejs\

Step 5:



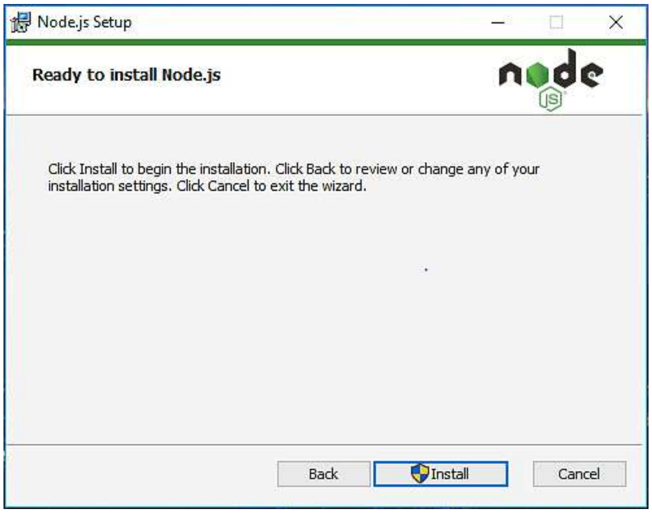
Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Install the core Node.js runtime (node.exe).

This feature requires 1635KB on your hard drive. It has 2 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.

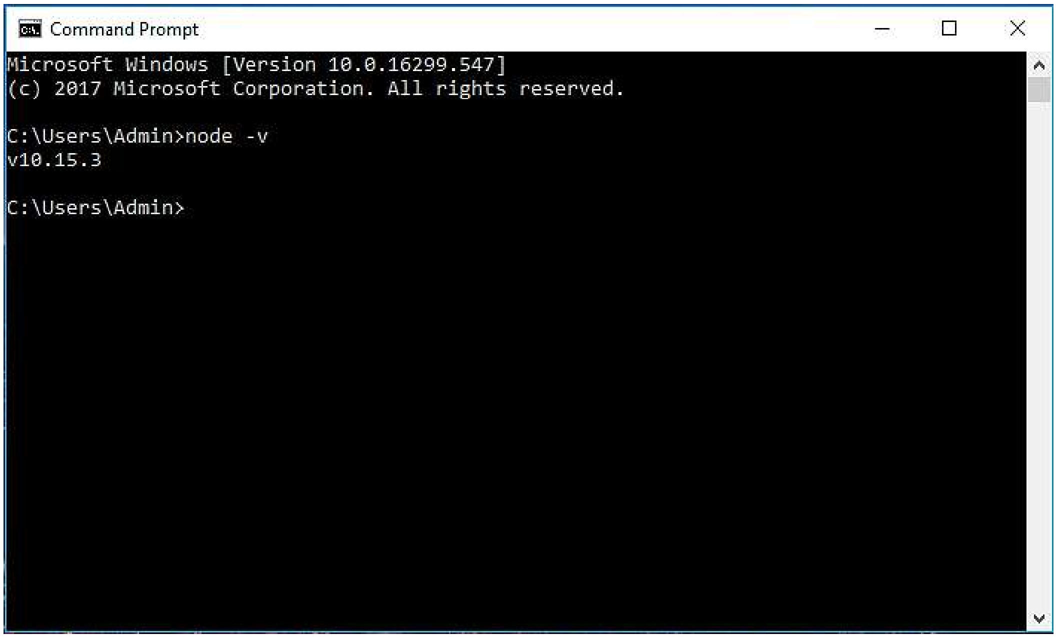


Ready to install Node.js

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Step 6:

C:\Users\Admin> node –v



```
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Admin>node -v
v10.15.3

C:\Users\Admin>
```

Results:

Thus, the above Program to Install Node.JS was installed successfully.

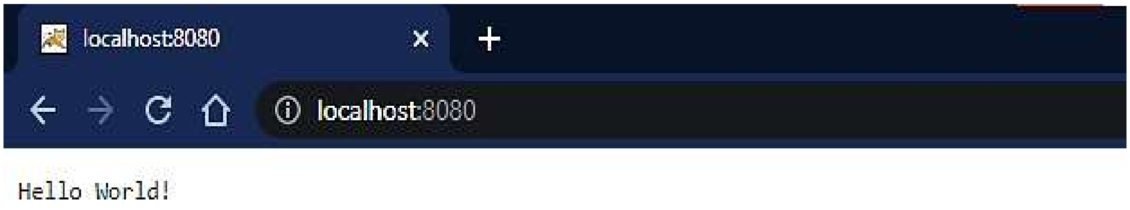
Create Web Server

Source Code:

httpWebServer.js

```
var http = require("http");
http
    .createServer(function (req, res) {
        res.write("Hello World!"); res.end();
    })
 .listen(8080);
```

Output:



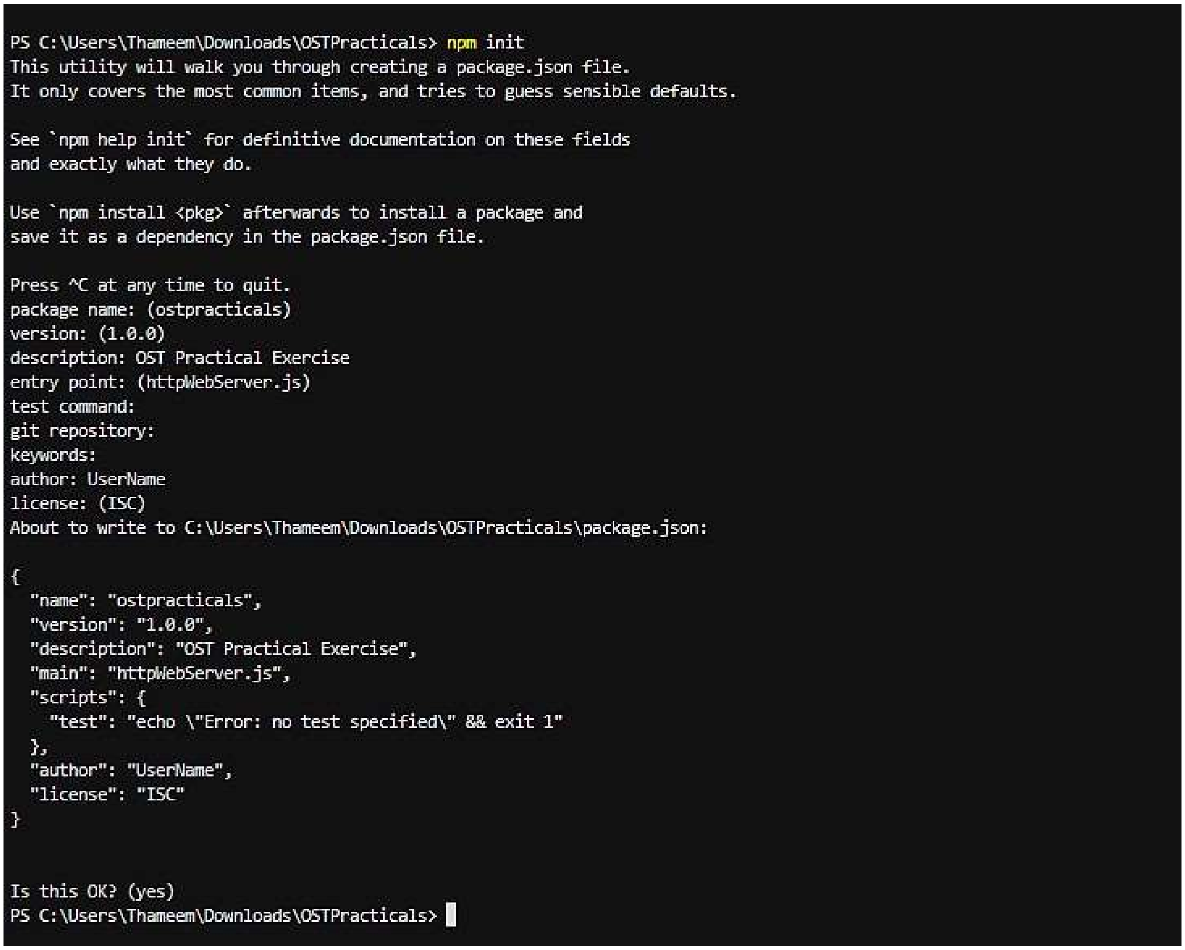
```
Hello World!
```

Results:

Thus, the above Program to Create Web Server was executed successfully.

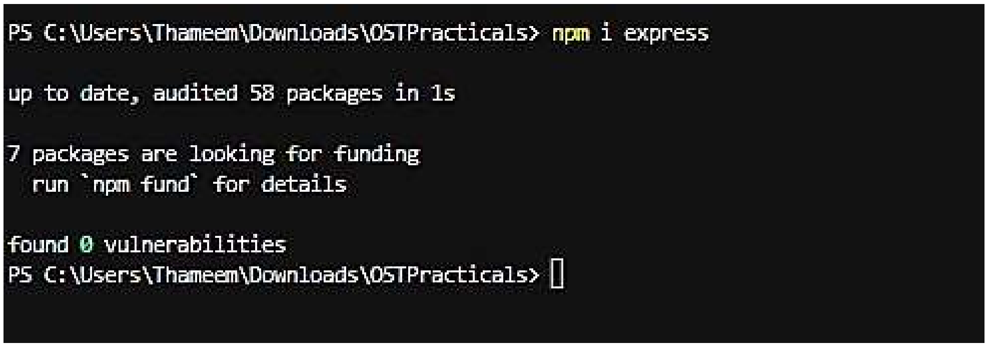
Use NPM Command

>>npm init



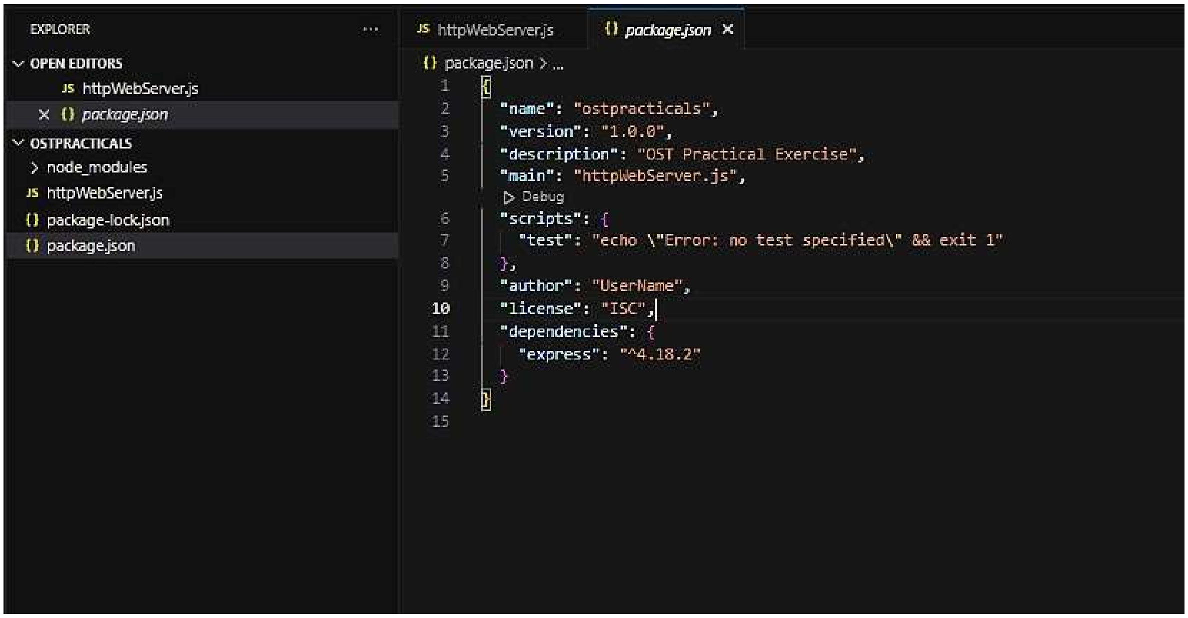
```
PS C:\Users\Thameem\Downloads\OSTPracticals> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

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: (ostpracticals)
version: (1.0.0)
description: OST Practical Exercise
entry point: (httpWebServer.js)
test command:
git repository:
keywords:
author: UserName
license: (ISC)
About to write to C:\Users\Thameem\Downloads\OSTPracticals\package.json:

{
  "name": "ostpracticals",
  "version": "1.0.0",
  "description": "OST Practical Exercise",
  "main": "httpWebServer.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "UserName",
  "license": "ISC"
}


Is this OK? (yes)
PS C:\Users\Thameem\Downloads\OSTPracticals>
```

>>npm i express



```
PS C:\Users\Thameem\Downloads\OSTPracticals> npm i express

up to date, audited 58 packages in 1s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Thameem\Downloads\OSTPracticals>
```

Output:



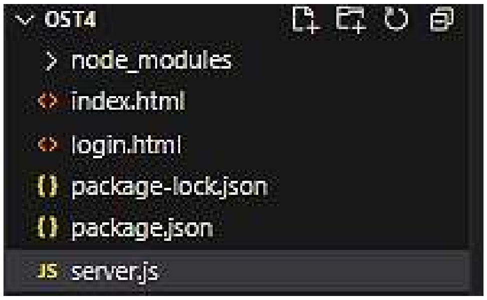
```json
{
  "name": "ostpracticals",
  "version": "1.0.0",
  "description": "OST Practical Exercise",
  "main": "httpWebServer.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "UserName",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2"
  }
}
```

Results:

Thus, the above Program to Use NPM Command was executed successfully.

Retrieve Values using Express

Folder Structure:



```
v OST4
  > node_modules
  <> index.html
  <> login.html
  {} package-lock.json
  {} package.json
  JS server.js
```

Source Code:

```
const express = require("express");

// Include ExpressJS
const app = express();

// Create an ExpressJS app
const bodyParser = require("body-parser");

// Middleware
app.use(bodyParser.urlencoded({ extended: false }));

// Route to Homepage
app.get("/", (req, res) => { res.sendFile(__dirname + "/index.html"); });

// Route to Login Page
app.get("/login", (req, res) => { res.sendFile(__dirname + "/login.html"); });
app.post("/login", (req, res) => {
  // Insert Login Code Here
  let username = req.body.username;let
  password = req.body.password;
  res.send(`Username: ${username} Password: ${password}`);
});
const port = 3000; // Port we will listen on
// Function to listen on the port
app.listen(port, () => console.log(`This app is listening on port ${port}`
));
```

index.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>Welcome Page</title>
  </head>
  <body>
    <a href="/login">Please Login Here!</a>
  </body>
</html>
```

login.html:

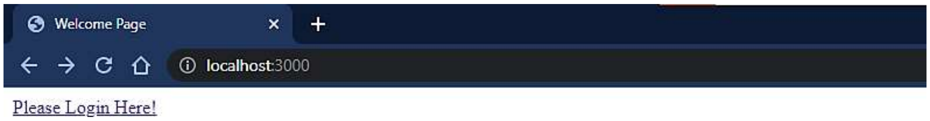
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Login Page</title>
  </head>
  <body>
    <form action="/login" method="post">
      <!-- user input-->
      Username:<br />
      <input
        type="text"
        name="username"
        placeholder="Username"
        required
      /><br /><br />
      Password:<br />
```

```
            <input
              type="password"
              name="password"
              placeholder="Password"
              required
            /><br /><br />
          <!-- submit button -->
          <input type="submit" value="login" />
        </form>
      </body>
    </html>
```

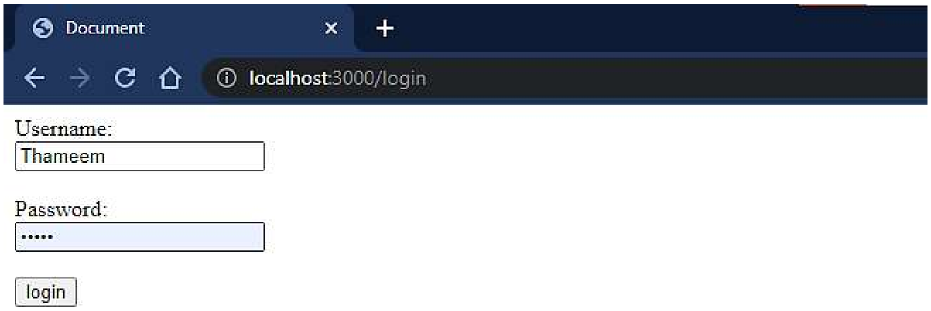
Package.json:

```
{
  "name": "ost4",
  "version": "1.0.0",
  "description": "OST practical Exercise 4","main":
  "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1","start": "node
    server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "express": "^4.18.2"
  }
}
```

Output:

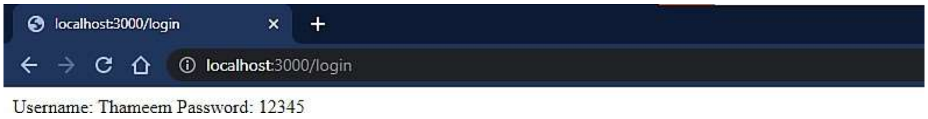


Please Login Here!



Username:
Thameem

Password:
•••••



```
Username: Thameem Password: 12345
```

Results:

Thus, the above Program to Retrieve Values using Express was executed successfully.

Highlight Menu using Angular.js

Source code:

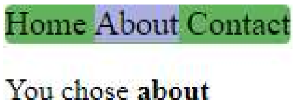
index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="style.css" rel="stylesheet" />
    <title>Document</title>
  </head>
  <body ng-app>
    <nav class="{{active}}" ng-click="$event.preventDefault()">
      <a href="#" class="home" ng-click="active='home'">Home</a>
      <a href="#" class="about" ng-click="active='about'">About</a>
      <a href="#" class="contact" ng-click="active='contact'">Contact</a>
    </nav>
    <p ng-show="active">You chose <b>{{active}}</b></p>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js"></script>
  </body>
</html>
```

style.css

```
a:hover, a:focus,
a:active {
  text-decoration: none;color:
  inherit;
  font-size: large;
}
nav {
  display: inline-block; background-
  color: #55b460;border-radius: 4px;
}
nav.home.home, nav.team.team,
nav.about.about,
nav.contact.contact {
  background-color: #a9aedd;
}
```

Output:



You chose **about**

Results:

Thus, the above Program to Highlight Menu using Angular.js was executed successfully.

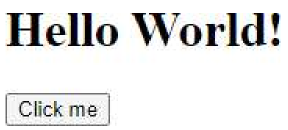
Manipulate Web Page using JQuery

Source Code:

index.html:

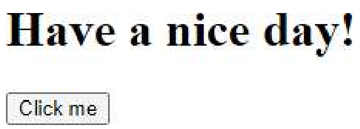
```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  </head>
  <body>
    <h1 id="myHeading">Hello World!</h1>
    <button onclick="changeText()">Click me</button>
    <script>
      function changeText() {
        $("#myHeading").text("Have a nice day!");
      }
    </script>
  </body>
</html>
```

Output:



Hello World!

After Button click:



Have a nice day!

Results:

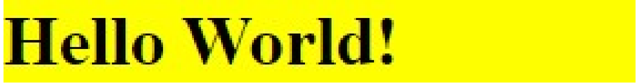
Thus, the above Program to Manipulate Web Page using JQuery was executed successfully.

Different Events of a Web Page using JQuery

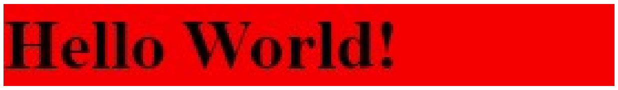
Source Code:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  </head>
  <body>
    <h1>Hello World!</h1>

       <script>
           $(document).ready(function () {
             $("h1").click(function () {
              $(this).hide();
        });
        $("h1").hover(function () {
          $(this).css("background-color", "yellow");
        });
        $("h1").mousedown(function () {
          $(this).css("background-color", "red");
        });
      });
    </script>
  </body>
</html>
```

Output:



```
Hello World!
```



```
Hello World!
```

Results:

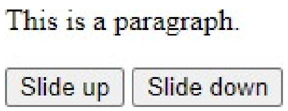
Thus, the above Program to Different Events of a Web Page using JQuery was executed successfully.

Different Effects using JQuery

Source Code:

```
<!DOCTYPE html>
<html>
    <head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
        <script>
            $(document).ready(function () {
                $(".btn1").click(function () {
                    $("p").slideUp();
                });
                $(".btn2").click(function () {
                    $("p").slideDown();
                });
            });
        </script>
    </head>
    <body>
        <p>This is a paragraph.</p>
        <button class="btn1">Slide up</button>
        <button class="btn2">Slide down</button>
    </body>
</html>
```

Output:



This is a paragraph.

Results:

Thus, the above Program to Different Effects using JQuery was executed successfully.