

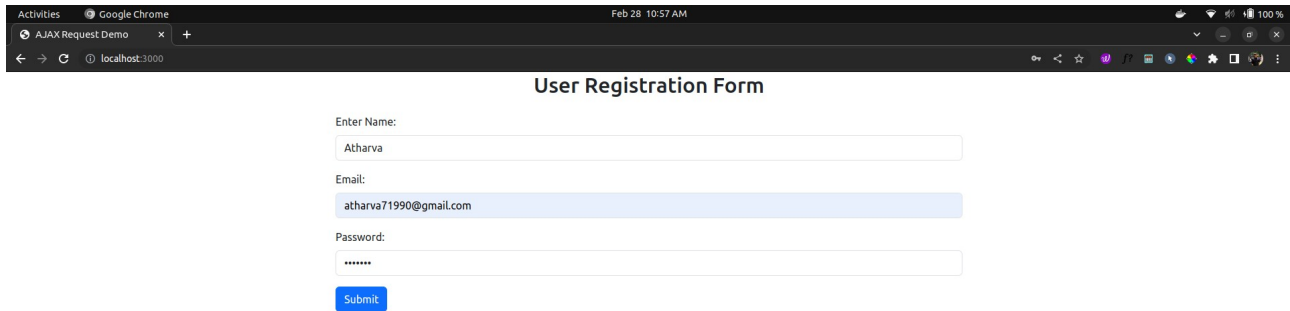
Assignment 1B: JavaScript AJAX Implementation

Name: Atharva Agrawal

RollNo: 33303

Batch: K-11

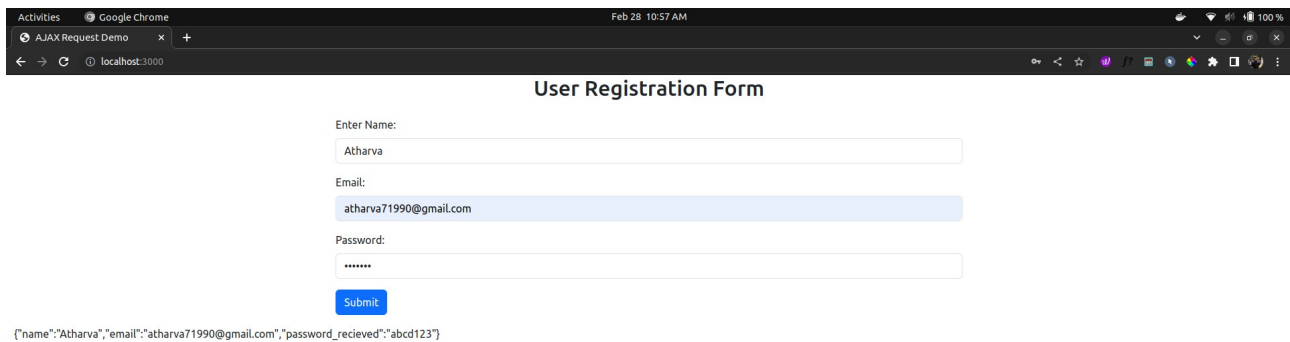
Sending Values:



The screenshot shows a Google Chrome browser window with the title "AJAX Request Demo" and the address bar showing "localhost:3000". The page displays a "User Registration Form" with the following fields and values:

- Enter Name: Atharva
- Email: atharva71990@gmail.com
- Password: *****
- Submit button

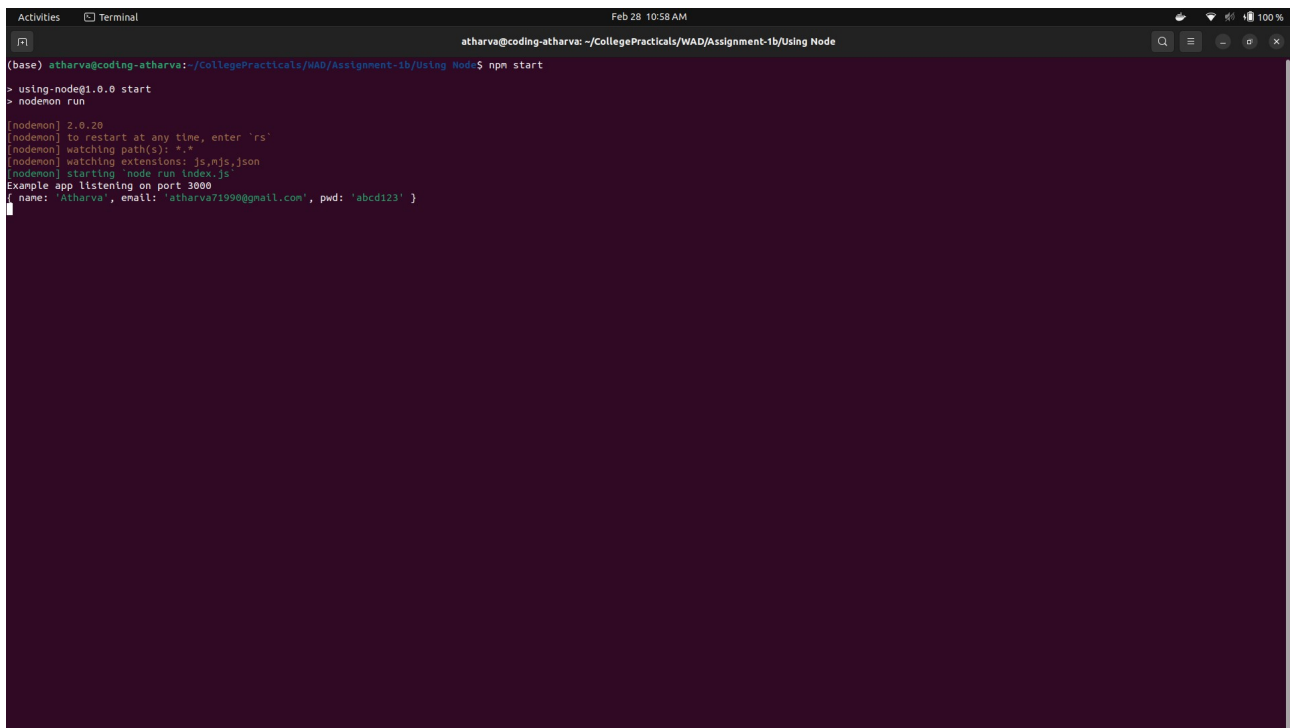
Received Data:



The screenshot shows the same Google Chrome browser window as above, but with a JSON response displayed in the console. The response is:

```
[{"name": "Atharva", "email": "atharva71990@gmail.com", "password_recieved": "abcd123"}]
```

Server Running:

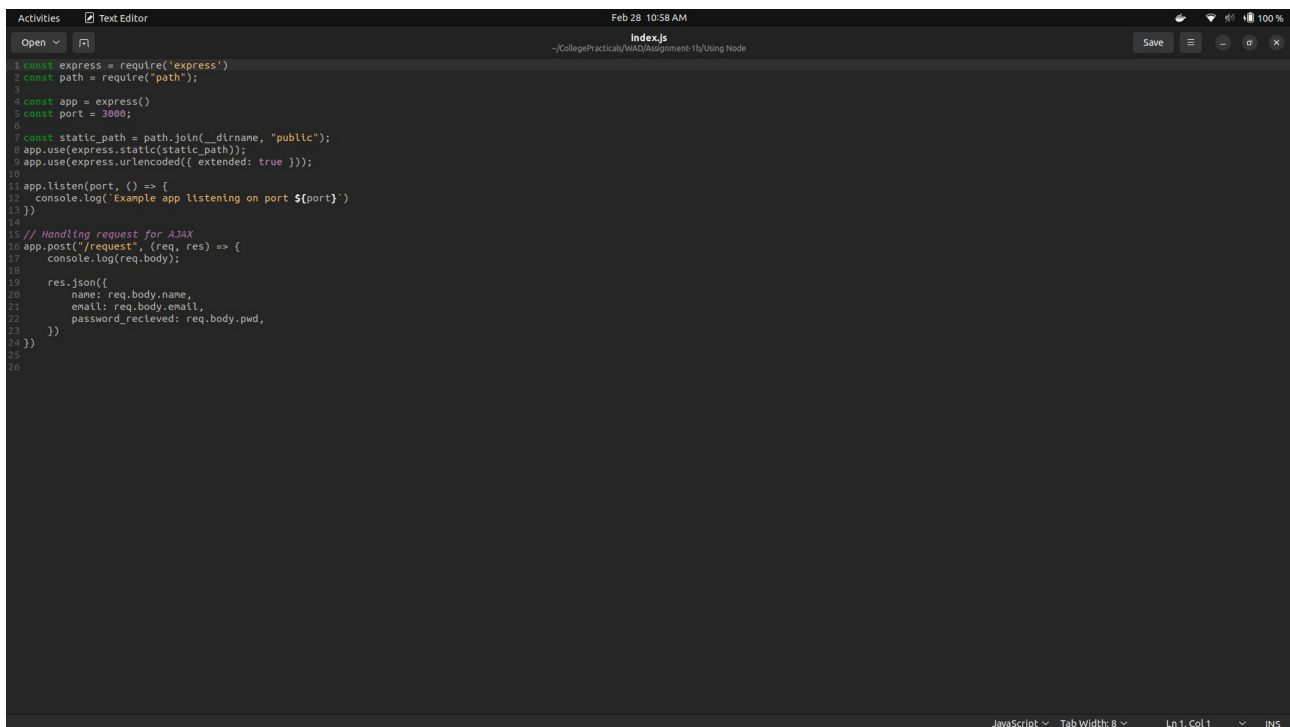


A terminal window titled 'atharva@coding-atharva: ~/CollegePracticals/WAD/Assignment-1b/Using Node' showing the process of starting a Node.js server. The user runs 'npm start', which triggers 'using-node@1.0.0 start' and 'nodemon run'. The output shows 'nodemon' version 2.0.20, instructions to restart with 'rs', and the server starting on port 3000. A sample JSON response is displayed: { name: 'Atharva', email: 'atharva71990@gmail.com', pwd: 'abcd123' }.

```
(base) atharva@coding-atharva:~/CollegePracticals/WAD/Assignment-1b/Using Node$ npm start
> using-node@1.0.0 start
> nodemon run

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node run index.js`
Example app listening on port 3000
{ name: 'Atharva', email: 'atharva71990@gmail.com', pwd: 'abcd123' }
```

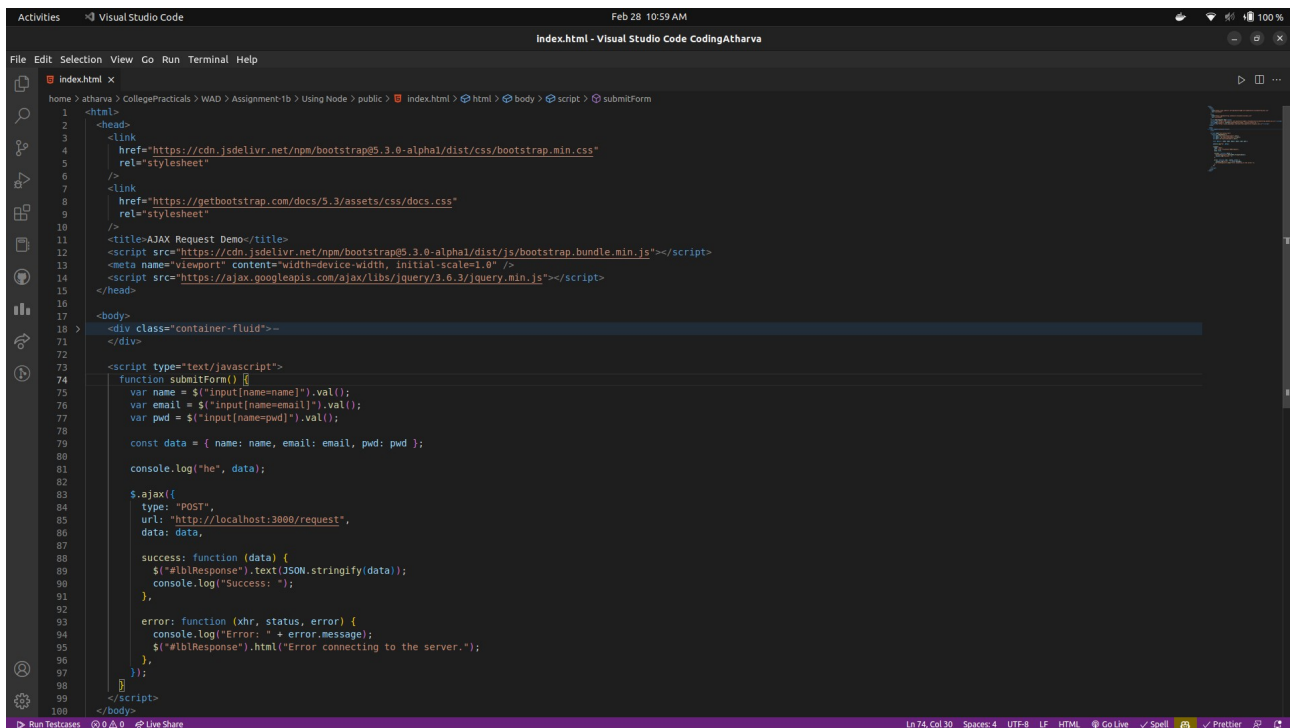
NodeJS Server Code:



A text editor window titled 'index.js' showing the code for a Node.js server. The code uses Express.js to create a server listening on port 3000. It handles POST requests to '/request' and returns a JSON response with the request body. The code is as follows:

```
1 const express = require('express')
2 const path = require('path')
3
4 const app = express()
5 const port = 3000;
6
7 const static_path = path.join(__dirname, "public");
8 app.use(express.static(static_path));
9 app.use(express.urlencoded({ extended: true }));
10
11 app.listen(port, () => {
12   console.log('Example app listening on port ${port}')
13 })
14
15 // Handling request for API
16 app.post("/request", (req, res) => {
17   console.log(req.body);
18
19   res.json({
20     name: req.body.name,
21     email: req.body.email,
22     password_retrieved: req.body.pwd,
23   })
24 })
25
26
```

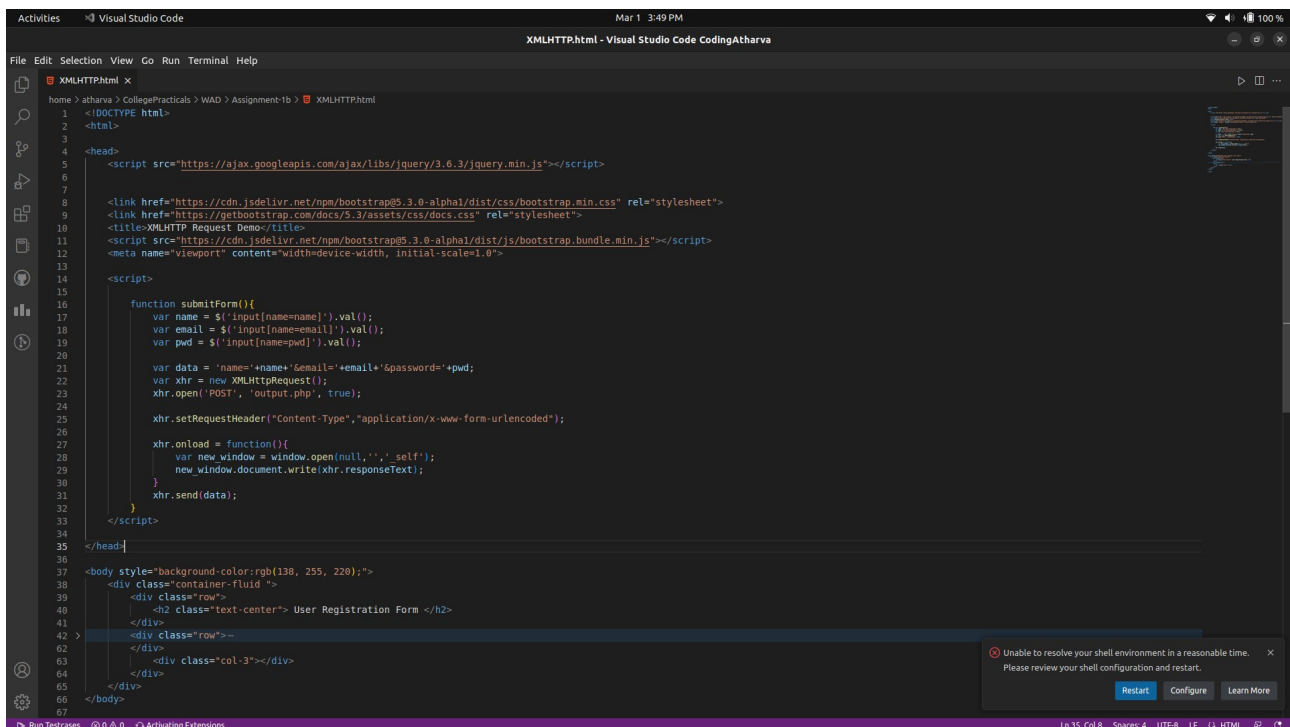
AJAX Code:



The screenshot shows the Visual Studio Code editor with the file 'index.html' open. The code is an HTML document with a title 'AJAX Request Demo'. It includes links to Bootstrap CSS and jQuery. The main content is a JavaScript function 'submitForm' that uses the jQuery \$.ajax() method to send a POST request to 'http://localhost:3000/request'. The success callback logs the response, and the error callback logs an error message.

```
1 <html>
2 <head>
3   <link
4     href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha/dist/css/bootstrap.min.css"
5     rel="stylesheet"
6   />
7   <link
8     href="https://getbootstrap.com/docs/5.3/assets/css/docs.css"
9     rel="stylesheet"
10  />
11  <title>AJAX Request Demo</title>
12  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha/dist/js/bootstrap.bundle.min.js"></script>
13  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
14  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
15 </head>
16
17 <body>
18   <div class="container-fluid">--
71   </div>
72
73   <script type="text/javascript">
74     function submitForm() {
75       var name = $('input[name=name]').val();
76       var email = $('input[name=email]').val();
77       var pwd = $('input[name=pwd]').val();
78
79       const data = { name: name, email: email, pwd: pwd };
80
81       console.log("he", data);
82
83       $.ajax({
84         type: "POST",
85         url: "http://localhost:3000/request",
86         data: data,
87
88         success: function (data) {
89           $('#lblResponse').text(JSON.stringify(data));
90           console.log("Success: ");
91         },
92
93         error: function (xhr, status, error) {
94           console.log("Error: " + error.message);
95           $('#lblResponse').html("Error connecting to the server.");
96         },
97       });
98     }
99   </script>
100 </body>
```

XMLHTTP Code:



The screenshot shows the Visual Studio Code editor with the file 'XMLHTTP.html' open. The code is an HTML document with a title 'XMLHTTP Request Demo'. It includes links to Bootstrap CSS and jQuery. The main content is a JavaScript function 'submitForm' that uses the XMLHttpRequest (XMLHttpRequest) object to send a POST request to 'http://localhost:3000/request'. The success callback logs the response, and the error callback logs an error message. A modal dialog is shown at the bottom right with the message 'Unable to resolve your shell environment in a reasonable time. Please review your shell configuration and restart.' with buttons for 'Restart', 'Configure', and 'Learn More'.

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
6
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha/dist/css/bootstrap.min.css" rel="stylesheet">
8   <link href="https://getbootstrap.com/docs/5.3/assets/css/docs.css" rel="stylesheet">
9   <title>XMLHTTP Request Demo</title>
10  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha/dist/js/bootstrap.bundle.min.js"></script>
11  <meta name="viewport" content="width=device-width, initial-scale=1.0">
12
13  <script>
14    function submitForm(){
15      var name = $('input[name=name]').val();
16      var email = $('input[name=email]').val();
17      var pwd = $('input[name=pwd]').val();
18
19      var data = 'name='+name+'&email='+email+'&password='+pwd;
20      var xhr = new XMLHttpRequest();
21      xhr.open('POST', 'output.php', true);
22
23      xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
24
25      xhr.onload = function(){
26        var new_window = window.open(null, '', 'self');
27        new_window.document.write(xhr.responseText);
28      }
29      xhr.send(data);
30    }
31  </script>
32
33 </head>
34
35 <body style="background-color:rgb(138, 255, 228);">
36   <div class="container-fluid">
37     <div class="row">
38       <div class="col">
39         <h2 class="text-center"> User Registration Form </h2>
40       </div>
41     </div>
42     <div class="row">--
62     <div class="col-3"></div>
63   </div>
64 </body>
65
66 </div>
```