

Assignment- 1

Name : Iwala Rohan KamleshBhai.

Roll No : 18.

Sem : 7th.

Subject : Application Development With Full Stack.(705).

Date : 30-Jul-23.

Github Link :- https://github.com/Rohaniwala/18_Rohan_iwala_PA1_705.

1. Develop a web server with following functionalities:

- Serve static resources.
- Handle GET request.
- Handle POST request.

Index.html

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

<head>
  <title>Title</title>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

  <!-- Bootstrap CSS v5.2.1 -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">

</head>

<body>
  <div class="container">
    <form action="/submit" method="post">
      <div class="mb-3">
        <label for="email" class="form-label">Email</label>
        <input type="email" class="form-control" name="email"
id="email" placeholder="Entar Email here ...">
      </div>
      <div class="mb-3">
        <label for="password" class="form-label">Password</label>
        <input type="password" class="form-control" name="password"
id="password"
          placeholder="Entar Password here ...">
      </div>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </div>
```

```

    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
    integrity="sha384-
oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
crossorigin="anonymous">
    </script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"
    integrity="sha384-
7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZX19NZ/e6oz"
crossorigin="anonymous">
    </script>
</body>

</html>

```

Index.js

```

const http = require("http");
const fs = require("fs");

http.createServer((req, res) => {
    const responseObject = { API: req.url, METHOD: req.method, Body:
(req.body) ? req.body : {} };
    if (req.url === '/' && req.method == 'GET') {
        res.setHeader("Content-Type", "application/json");
        res.write(JSON.stringify(responseObject));
        res.end();
    }
    else if (req.url === '/data' && req.method == 'GET') {
        res.setHeader("Content-Type", "application/json");
        res.write(JSON.stringify(responseObject));
        res.end();
    }
    else if (req.url === '/form' && req.method == 'GET') {
        fs.readFile('./index.html', 'utf8', (err, data) => {
            if (err) {
                res.writeHead(500, { 'Content-Type': 'text/plain' });
                res.end('Internal Server Error');
            } else {
                res.writeHead(200, { 'Content-Type': 'text/html' });
                res.end(data);
            }
        });
    }
});

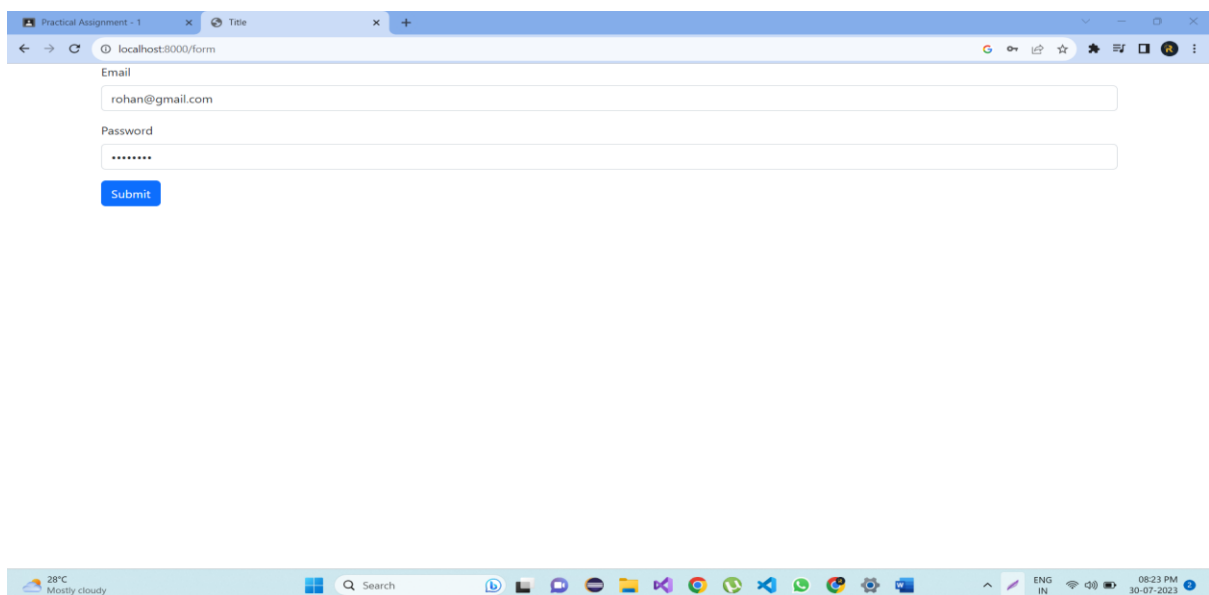
```

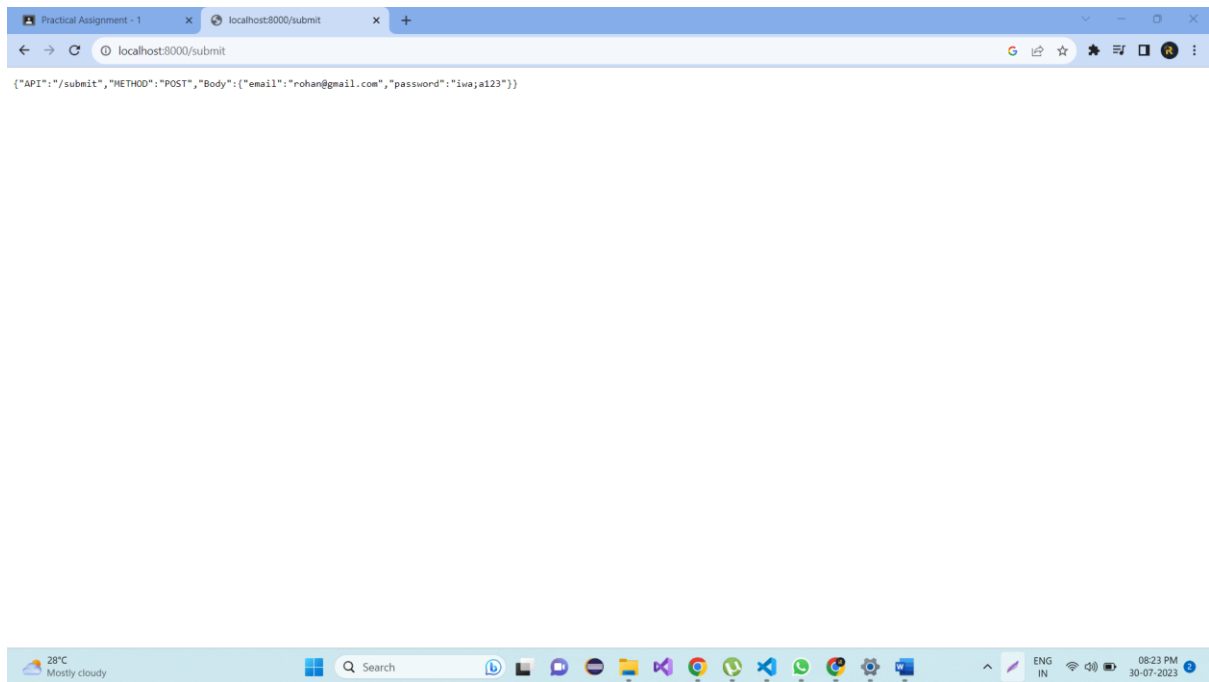
```

}
else if (req.url === '/submit' && req.method === 'POST') {
  let body = '';
  req.on('data', (chunk) => {
    body += chunk.toString();
  });
  req.on('end', () => {
    const formData = new URLSearchParams(body);
    const email = formData.get('email');
    const password = formData.get('password');
    responseObject.Body = {
      email: email,
      password: password
    }
    res.write(JSON.stringify(responseObject));
    res.end();
  });
}
else {
  res.write(JSON.stringify(responseObject));
  res.end();
}
}).listen(8000, () => {
  console.log("server is listening on port 8000");
  console.log("localhost:8000/");
});

```

Output:





2. Develop nodejs application with following requirements:

- Develop a route "/gethello" with GET method. It displays "Hello NodeJS!!" as response.
- Make an HTML page and display.
- Call "/gethello" route from HTML page using AJAX call. (Any frontend AJAX call API can be used.)

Ajaxcall.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>
  <div id="content-Page">

    </div>
    <button onclick="fetchdata()">Fecth Data</button>
  </body>
</html>
```

```

</body>
<script>
    function fetchdata() {
        var httpRequest = new XMLHttpRequest();
        httpRequest.onreadystatechange = function () {
            if (httpRequest.readyState == 4 && httpRequest.status == 200) {
                document.getElementById("content-Page").innerHTML =
httpRequest.responseText
            }
        };
        httpRequest.open("GET", '/gethello', true);
        httpRequest.send();
    }
</script>

</html>

```

Hello.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>
    <h1>hello Rohan IWala</h1>
</body>

</html>

```

Server.js

```

var http = require('http');
var fs = require('fs');
http.createServer((req, res) => {
    if (req.method === 'GET' && req.url === '/') {
        res.end("hello Page 1");
    }

    if (req.url === '/gethello') {
        fs.readFile('./hello.html', (err, data) => {
            if (err) {
                fs.write("page not found");
            }
        });
    }
});

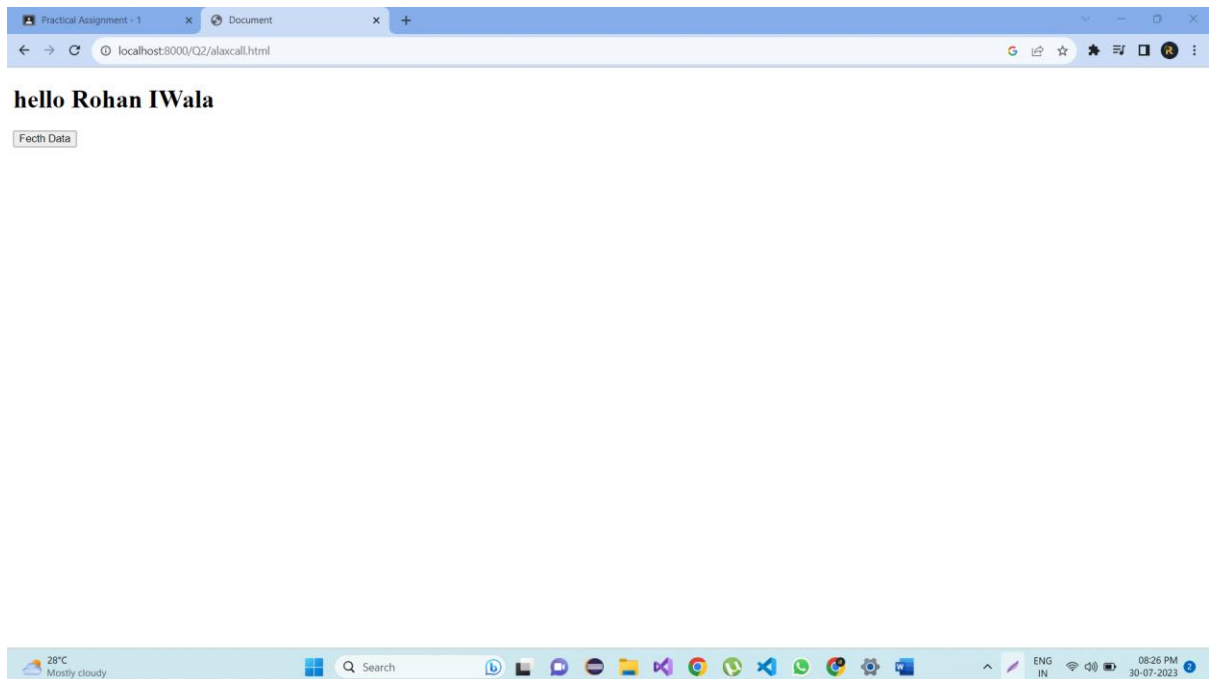
```

```

        fs.end();
    }
    else {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write(data);
        res.end();
    }
})
}
if (req.url === '/Q2/alaxcall.html') {
    fs.readFile('./alaxcall.html', (err, data) => {
        if (err) {
            fs.write("pagenotfound");
            fs.end();
        }
        else {
            res.writeHead(200, { 'Content-Type': 'text/html' });
            res.write(data);
            res.end();
        }
    })
}
// console.log("hello");
}).listen(8000);

```

Output:



3. Develop a module for domain specific chatbot and use it in a command line application.

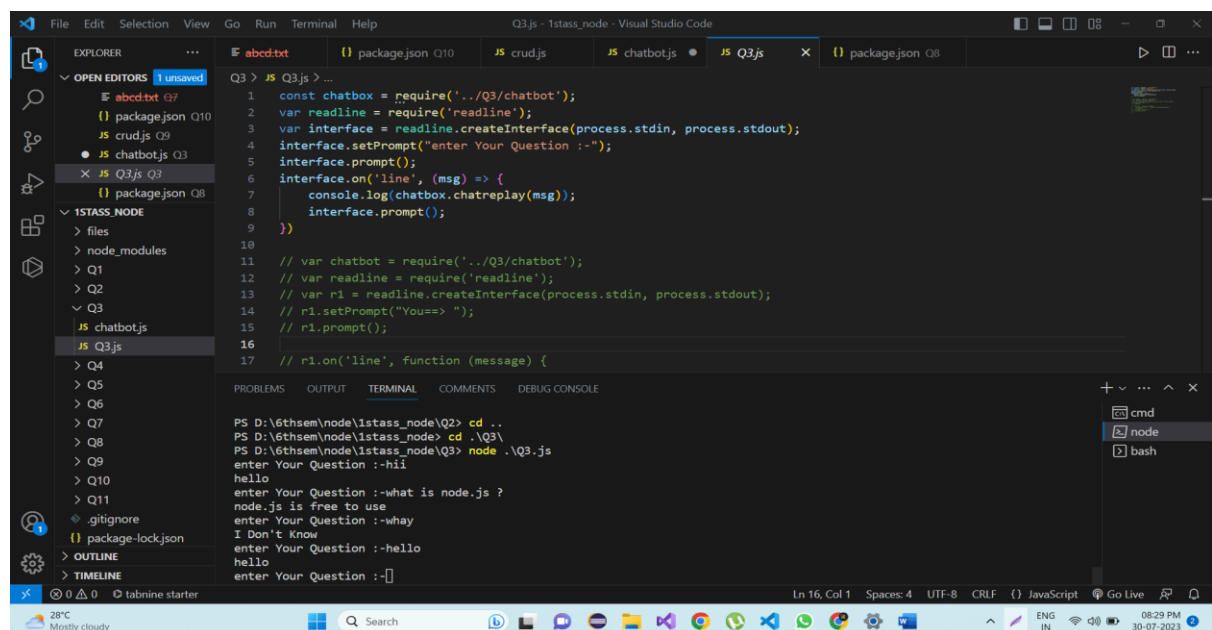
Chatbot.js

```
module.exports.chatreply = (message) => {  
  if (message.toLowerCase().indexOf('hii') > -1 ||  
message.toLowerCase().indexOf('hello') > -1) {  
    return "hello";  
  }  
  else if (message.toLowerCase().indexOf('What is Node.js ?') > -1 ||  
message.toLowerCase().indexOf('node.js') > -1) {  
    return "node.js is free to use";  
  }  
  else {  
    return "I Don't Know";  
  }  
}
```

Q3.js

```
const chatbox = require('../Q3/chatbot');  
var readline = require('readline');  
var interface = readline.createInterface(process.stdin, process.stdout);  
interface.setPrompt("enter Your Question :-");  
interface.prompt();  
interface.on('line', (msg) => {  
  console.log(chatbox.chatreply(msg));  
  interface.prompt();  
})
```

Output:



The screenshot shows the Visual Studio Code editor with a project named 'Q3.js - istass_node'. The Explorer sidebar on the left shows the file structure with 'Q3.js' selected. The main editor displays the code for 'Q3.js', which imports the 'chatbot' module and uses the 'readline' module to create an interactive interface. The interface prompts the user to enter a question and logs the response from the 'chatbot' module. The Terminal at the bottom shows the execution of the program, with the user entering 'hii', 'hello', 'what is node.js ?', 'node.js is free to use', 'why', 'I Don't Know', and 'hello', all of which are correctly handled by the chatbot module.

```
Q3 > JS Q3.js > ...  
1  const chatbox = require('../Q3/chatbot');  
2  var readline = require('readline');  
3  var interface = readline.createInterface(process.stdin, process.stdout);  
4  interface.setPrompt("enter Your Question :-");  
5  interface.prompt();  
6  interface.on('line', (msg) => {  
7      console.log(chatbox.chatreply(msg));  
8      interface.prompt();  
9  })  
10  
11  // var chatbot = require('../Q3/chatbot');  
12  // var readline = require('readline');  
13  // var rl = readline.createInterface(process.stdin, process.stdout);  
14  // rl.setPrompt("You==> ");  
15  // rl.prompt();  
16  
17  // rl.on('line', function (message) {  
18  
19  })  
20  
21  // rl.on('line', function (message) {  
22  
23  })  
24  
25  // rl.on('line', function (message) {  
26  
27  })  
28  
29  // rl.on('line', function (message) {  
30  
31  })  
32  
33  // rl.on('line', function (message) {  
34  
35  })  
36  
37  // rl.on('line', function (message) {  
38  
39  })  
40  
41  // rl.on('line', function (message) {  
42  
43  })  
44  
45  // rl.on('line', function (message) {  
46  
47  })  
48  
49  // rl.on('line', function (message) {  
50  
51  })  
52  
53  // rl.on('line', function (message) {  
54  
55  })  
56  
57  // rl.on('line', function (message) {  
58  
59  })  
60  
61  // rl.on('line', function (message) {  
62  
63  })  
64  
65  // rl.on('line', function (message) {  
66  
67  })  
68  
69  // rl.on('line', function (message) {  
70  
71  })  
72  
73  // rl.on('line', function (message) {  
74  
75  })  
76  
77  // rl.on('line', function (message) {  
78  
79  })  
80  
81  // rl.on('line', function (message) {  
82  
83  })  
84  
85  // rl.on('line', function (message) {  
86  
87  })  
88  
89  // rl.on('line', function (message) {  
90  
91  })  
92  
93  // rl.on('line', function (message) {  
94  
95  })  
96  
97  // rl.on('line', function (message) {  
98  
99  })  
100  
101  // rl.on('line', function (message) {  
102  
103  })  
104  
105  // rl.on('line', function (message) {  
106  
107  })  
108  
109  // rl.on('line', function (message) {  
110  
111  })  
112  
113  // rl.on('line', function (message) {  
114  
115  })  
116  
117  // rl.on('line', function (message) {  
118  
119  })  
120  
121  // rl.on('line', function (message) {  
122  
123  })  
124  
125  // rl.on('line', function (message) {  
126  
127  })  
128  
129  // rl.on('line', function (message) {  
130  
131  })  
132  
133  // rl.on('line', function (message) {  
134  
135  })  
136  
137  // rl.on('line', function (message) {  
138  
139  })  
140  
141  // rl.on('line', function (message) {  
142  
143  })  
144  
145  // rl.on('line', function (message) {  
146  
147  })  
148  
149  // rl.on('line', function (message) {  
150  
151  })  
152  
153  // rl.on('line', function (message) {  
154  
155  })  
156  
157  // rl.on('line', function (message) {  
158  
159  })  
160  
161  // rl.on('line', function (message) {  
162  
163  })  
164  
165  // rl.on('line', function (message) {  
166  
167  })  
168  
169  // rl.on('line', function (message) {  
170  
171  })  
172  
173  // rl.on('line', function (message) {  
174  
175  })  
176  
177  // rl.on('line', function (message) {  
178  
179  })  
180  
181  // rl.on('line', function (message) {  
182  
183  })  
184  
185  // rl.on('line', function (message) {  
186  
187  })  
188  
189  // rl.on('line', function (message) {  
190  
191  })  
192  
193  // rl.on('line', function (message) {  
194  
195  })  
196  
197  // rl.on('line', function (message) {  
198  
199  })  
200  
201  // rl.on('line', function (message) {  
202  
203  })  
204  
205  // rl.on('line', function (message) {  
206  
207  })  
208  
209  // rl.on('line', function (message) {  
210  
211  })  
212  
213  // rl.on('line', function (message) {  
214  
215  })  
216  
217  // rl.on('line', function (message) {  
218  
219  })  
220  
221  // rl.on('line', function (message) {  
222  
223  })  
224  
225  // rl.on('line', function (message) {  
226  
227  })  
228  
229  // rl.on('line', function (message) {  
230  
231  })  
232  
233  // rl.on('line', function (message) {  
234  
235  })  
236  
237  // rl.on('line', function (message) {  
238  
239  })  
240  
241  // rl.on('line', function (message) {  
242  
243  })  
244  
245  // rl.on('line', function (message) {  
246  
247  })  
248  
249  // rl.on('line', function (message) {  
250  
251  })  
252  
253  // rl.on('line', function (message) {  
254  
255  })  
256  
257  // rl.on('line', function (message) {  
258  
259  })  
260  
261  // rl.on('line', function (message) {  
262  
263  })  
264  
265  // rl.on('line', function (message) {  
266  
267  })  
268  
269  // rl.on('line', function (message) {  
270  
271  })  
272  
273  // rl.on('line', function (message) {  
274  
275  })  
276  
277  // rl.on('line', function (message) {  
278  
279  })  
280  
281  // rl.on('line', function (message) {  
282  
283  })  
284  
285  // rl.on('line', function (message) {  
286  
287  })  
288  
289  // rl.on('line', function (message) {  
290  
291  })  
292  
293  // rl.on('line', function (message) {  
294  
295  })  
296  
297  // rl.on('line', function (message) {  
298  
299  })  
300  
301  // rl.on('line', function (message) {  
302  
303  })  
304  
305  // rl.on('line', function (message) {  
306  
307  })  
308  
309  // rl.on('line', function (message) {  
310  
311  })  
312  
313  // rl.on('line', function (message) {  
314  
315  })  
316  
317  // rl.on('line', function (message) {  
318  
319  })  
320  
321  // rl.on('line', function (message) {  
322  
323  })  
324  
325  // rl.on('line', function (message) {  
326  
327  })  
328  
329  // rl.on('line', function (message) {  
330  
331  })  
332  
333  // rl.on('line', function (message) {  
334  
335  })  
336  
337  // rl.on('line', function (message) {  
338  
339  })  
340  
341  // rl.on('line', function (message) {  
342  
343  })  
344  
345  // rl.on('line', function (message) {  
346  
347  })  
348  
349  // rl.on('line', function (message) {  
350  
351  })  
352  
353  // rl.on('line', function (message) {  
354  
355  })  
356  
357  // rl.on('line', function (message) {  
358  
359  })  
360  
361  // rl.on('line', function (message) {  
362  
363  })  
364  
365  // rl.on('line', function (message) {  
366  
367  })  
368  
369  // rl.on('line', function (message) {  
370  
371  })  
372  
373  // rl.on('line', function (message) {  
374  
375  })  
376  
377  // rl.on('line', function (message) {  
378  
379  })  
380  
381  // rl.on('line', function (message) {  
382  
383  })  
384  
385  // rl.on('line', function (message) {  
386  
387  })  
388  
389  // rl.on('line', function (message) {  
390  
391  })  
392  
393  // rl.on('line', function (message) {  
394  
395  })  
396  
397  // rl.on('line', function (message) {  
398  
399  })  
400  
401  // rl.on('line', function (message) {  
402  
403  })  
404  
405  // rl.on('line', function (message) {  
406  
407  })  
408  
409  // rl.on('line', function (message) {  
410  
411  })  
412  
413  // rl.on('line', function (message) {  
414  
415  })  
416  
417  // rl.on('line', function (message) {  
418  
419  })  
420  
421  // rl.on('line', function (message) {  
422  
423  })  
424  
425  // rl.on('line', function (message) {  
426  
427  })  
428  
429  // rl.on('line', function (message) {  
430  
431  })  
432  
433  // rl.on('line', function (message) {  
434  
435  })  
436  
437  // rl.on('line', function (message) {  
438  
439  })  
440  
441  // rl.on('line', function (message) {  
442  
443  })  
444  
445  // rl.on('line', function (message) {  
446  
447  })  
448  
449  // rl.on('line', function (message) {  
450  
451  })  
452  
453  // rl.on('line', function (message) {  
454  
455  })  
456  
457  // rl.on('line', function (message) {  
458  
459  })  
460  
461  // rl.on('line', function (message) {  
462  
463  })  
464  
465  // rl.on('line', function (message) {  
466  
467  })  
468  
469  // rl.on('line', function (message) {  
470  
471  })  
472  
473  // rl.on('line', function (message) {  
474  
475  })  
476  
477  // rl.on('line', function (message) {  
478  
479  })  
480  
481  // rl.on('line', function (message) {  
482  
483  })  
484  
485  // rl.on('line', function (message) {  
486  
487  })  
488  
489  // rl.on('line', function (message) {  
490  
491  })  
492  
493  // rl.on('line', function (message) {  
494  
495  })  
496  
497  // rl.on('line', function (message) {  
498  
499  })  
500  
501  // rl.on('line', function (message) {  
502  
503  })  
504  
505  // rl.on('line', function (message) {  
506  
507  })  
508  
509  // rl.on('line', function (message) {  
510  
511  })  
512  
513  // rl.on('line', function (message) {  
514  
515  })  
516  
517  // rl.on('line', function (message) {  
518  
519  })  
520  
521  // rl.on('line', function (message) {  
522  
523  })  
524  
525  // rl.on('line', function (message) {  
526  
527  })  
528  
529  // rl.on('line', function (message) {  
530  
531  })  
532  
533  // rl.on('line', function (message) {  
534  
535  })  
536  
537  // rl.on('line', function (message) {  
538  
539  })  
540  
541  // rl.on('line', function (message) {  
542  
543  })  
544  
545  // rl.on('line', function (message) {  
546  
547  })  
548  
549  // rl.on('line', function (message) {  
550  
551  })  
552  
553  // rl.on('line', function (message) {  
554  
555  })  
556  
557  // rl.on('line', function (message) {  
558  
559  })  
560  
561  // rl.on('line', function (message) {  
562  
563  })  
564  
565  // rl.on('line', function (message) {  
566  
567  })  
568  
569  // rl.on('line', function (message) {  
570  
571  })  
572  
573  // rl.on('line', function (message) {  
574  
575  })  
576  
577  // rl.on('line', function (message) {  
578  
579  })  
580  
581  // rl.on('line', function (message) {  
582  
583  })  
584  
585  // rl.on('line', function (message) {  
586  
587  })  
588  
589  // rl.on('line', function (message) {  
590  
591  })  
592  
593  // rl.on('line', function (message) {  
594  
595  })  
596  
597  // rl.on('line', function (message) {  
598  
599  })  
600  
601  // rl.on('line', function (message) {  
602  
603  })  
604  
605  // rl.on('line', function (message) {  
606  
607  })  
608  
609  // rl.on('line', function (message) {  
610  
611  })  
612  
613  // rl.on('line', function (message) {  
614  
615  })  
616  
617  // rl.on('line', function (message) {  
618  
619  })  
620  
621  // rl.on('line', function (message) {  
622  
623  })  
624  
625  // rl.on('line', function (message) {  
626  
627  })  
628  
629  // rl.on('line', function (message) {  
630  
631  })  
632  
633  // rl.on('line', function (message) {  
634  
635  })  
636  
637  // rl.on('line', function (message) {  
638  
639  })  
640  
641  // rl.on('line', function (message) {  
642  
643  })  
644  
645  // rl.on('line', function (message) {  
646  
647  })  
648  
649  // rl.on('line', function (message) {  
650  
651  })  
652  
653  // rl.on('line', function (message) {  
654  
655  })  
656  
657  // rl.on('line', function (message) {  
658  
659  })  
660  
661  // rl.on('line', function (message) {  
662  
663  })  
664  
665  // rl.on('line', function (message) {  
666  
667  })  
668  
669  // rl.on('line', function (message) {  
670  
671  })  
672  
673  // rl.on('line', function (message) {  
674  
675  })  
676  
677  // rl.on('line', function (message) {  
678  
679  })  
680  
681  // rl.on('line', function (message) {  
682  
683  })  
684  
685  // rl.on('line', function (message) {  
686  
687  })  
688  
689  // rl.on('line', function (message) {  
690  
691  })  
692  
693  // rl.on('line', function (message) {  
694  
695  })  
696  
697  // rl.on('line', function (message) {  
698  
699  })  
700  
701  // rl.on('line', function (message) {  
702  
703  })  
704  
705  // rl.on('line', function (message) {  
706  
707  })  
708  
709  // rl.on('line', function (message) {  
710  
711  })  
712  
713  // rl.on('line', function (message) {  
714  
715  })  
716  
717  // rl.on('line', function (message) {  
718  
719  })  
720  
721  // rl.on('line', function (message) {  
722  
723  })  
724  
725  // rl.on('line', function (message) {  
726  
727  })  
728  
729  // rl.on('line', function (message) {  
730  
731  })  
732  
733  // rl.on('line', function (message) {  
734  
735  })  
736  
737  // rl.on('line', function (message) {  
738  
739  })  
740  
741  // rl.on('line', function (message) {  
742  
743  })  
744  
745  // rl.on('line', function (message) {  
746  
747  })  
748  
749  // rl.on('line', function (message) {  
750  
751  })  
752  
753  // rl.on('line', function (message) {  
754  
755  })  
756  
757  // rl.on('line', function (message) {  
758  
759  })  
760  
761  // rl.on('line', function (message) {  
762  
763  })  
764  
765  // rl.on('line', function (message) {  
766  
767  })  
768  
769  // rl.on('line', function (message) {  
770  
771  })  
772  
773  // rl.on('line', function (message) {  
774  
775  })  
776  
777  // rl.on('line', function (message) {  
778  
779  })  
780  
781  // rl.on('line', function (message) {  
782  
783  })  
784  
785  // rl.on('line', function (message) {  
786  
787  })  
788  
789  // rl.on('line', function (message) {  
790  
791  })  
792  
793  // rl.on('line', function (message) {  
794  
795  })  
796  
797  // rl.on('line', function (message) {  
798  
799  })  
800  
801  // rl.on('line', function (message) {  
802  
803  })  
804  
805  // rl.on('line', function (message) {  
806  
807  })  
808  
809  // rl.on('line', function (message) {  
810  
811  })  
812  
813  // rl.on('line', function (message) {  
814  
815  })  
816  
817  // rl.on('line', function (message) {  
818  
819  })  
820  
821  // rl.on('line', function (message) {  
822  
823  })  
824  
825  // rl.on('line', function (message) {  
826  
827  })  
828  
829  // rl.on('line', function (message) {  
830  
831  })  
832  
833  // rl.on('line', function (message) {  
834  
835  })  
836  
837  // rl.on('line', function (message) {  
838  
839  })  
840  
841  // rl.on('line', function (message) {  
842  
843  })  
844  
845  // rl.on('line', function (message) {  
846  
847  })  
848  
849  // rl.on('line', function (message) {  
850  
851  })  
852  
853  // rl.on('line', function (message) {  
854  
855  })  
856  
857  // rl.on('line', function (message) {  
858  
859  })  
860  
861  // rl.on('line', function (message) {  
862  
863  })  
864  
865  // rl.on('line', function (message) {  
866  
867  })  
868  
869  // rl.on('line', function (message) {  
870  
871  })  
872  
873  // rl.on('line', function (message) {  
874  
875  })  
876  
877  // rl.on('line', function (message) {  
878  
879  })  
880  
881  // rl.on('line', function (message) {  
882  
883  })  
884  
885  // rl.on('line', function (message) {  
886  
887  })  
888  
889  // rl.on('line', function (message) {  
890  
891  })  
892  
893  // rl.on('line', function (message) {  
894  
895  })  
896  
897  // rl.on('line', function (message) {  
898  
899  })  
900  
901  // rl.on('line', function (message) {  
902  
903  })  
904  
905  // rl.on('line', function (message) {  
906  
907  })  
908  
909  // rl.on('line', function (message) {  
910  
911  })  
912  
913  // rl.on('line', function (message) {  
914  
915  })  
916  
917  // rl.on('line', function (message) {  
918  
919  })  
920  
921  // rl.on('line', function (message) {  
922  
923  })  
924  
925  // rl.on('line', function (message) {  
926  
927  })  
928  
929  // rl.on('line', function (message) {  
930  
931  })  
932  
933  // rl.on('line', function (message) {  
934  
935  })  
936  
937  // rl.on('line', function (message) {  
938  
939  })  
940  
941  // rl.on('line', function (message) {  
942  
943  })  
944  
945  // rl.on('line', function (message) {  
946  
947  })  
948  
949  // rl.on('line', function (message) {  
950  
951  })  
952  
953  // rl.on('line', function (message) {  
954  
955  })  
956  
957  // rl.on('line', function (message) {  
958  
959  })  
960  
961  // rl.on('line', function (message) {  
962  
963  })  
964  
965  // rl.on('line', function (message) {  
966  
967  })  
968  
969  // rl.on('line', function (message) {  
970  
971  })  
972  
973  // rl.on('line', function (message) {  
974  
975  })  
976  
977  // rl.on('line', function (message) {  
978  
979  })  
980  
981  // rl.on('line', function (message) {  
982  
983  })  
984  
985  // rl.on('line', function (message) {  
986  
987  })  
988  
989  // rl.on('line', function (message) {  
990  
991  })  
992  
993  // rl.on('line', function (message) {  
994  
995  })  
996  
997  // rl.on('line', function (message) {  
998  
999  })  
1000  
1001  // rl.on('line', function (message) {  
1002  
1003  })  
1004  
1005  // rl.on('line', function (message) {  
1006  
1007  })  
1008  
1009  // rl.on('line', function (message) {  
1010  
1011  })  
1012  
1013  // rl.on('line', function (message) {  
1014  
1015  })  
1016  
1017  // rl.on('line', function (message) {  
1018  
1019  })  
1020  
1021  // rl.on('line', function (message) {  
1022  
1023  })  
1024  
1025  // rl.on('line', function (message) {  
1026  
1027  })  
1028  
1029  // rl.on('line', function (message) {  
1030  
1031  })  
1032  
1033  // rl.on('line', function (message) {  
1034  
1035  })  
1036  
1037  // rl.on('line', function (message) {  
1038  
1039  })  
1040  
1041  // rl.on('line', function (message) {  
1042  
1043  })  
1044  
1045  // rl.on('line', function (message) {  
1046  
1047  })  
1048  
1049  // rl.on('line', function (message) {  
1050  
1051  })  
1052  
1053  // rl.on('line', function (message) {  
1054  
1055  })  
1056  
1057  // rl.on('line', function (message) {  
1058  
1059  })  
1060  
1061  // rl.on('line', function (message) {  
1062  
1063  })  
1064  
1065  // rl.on('line', function (message) {  
1066  
1067  })  
1068  
1069  // rl.on('line', function (message) {  
1070  
1071  })  
1072  
1073  // rl.on('line', function (message) {  
1074  
1075  })  
1076  
1077  // rl.on('line', function (message) {  
1078  
1079  })  
1080  
1081  // rl.on('line', function (message) {  
1082  
1083  })  
1084  
1085  // rl.on('line', function (message) {  
1086  
1087  })  
1088  
1089  // rl.on('line', function (message) {  
1090  
1091  })  
1092  
1093  // rl.on('line', function (message) {  
1094  
1095  })  
1096  
1097  // rl.on('line', function (message) {  
1098  
1099  })  
1100  
1101  // rl.on('line', function (message) {  
1102  
1103  })  
1104  
1105  // rl.on('line', function (message) {  
1106  
1107  })  
1108  
1109  // rl.on('line', function (message) {  
1110  
1111  })  
1112  
1113  // rl.on('line', function (message) {  
1114  
1115  })  
1116  
1117  // rl.on('line', function (message) {  
1118  
1119  })  
1120  
1121  // rl.on('line', function (message) {  
1122  
1123  })  
1124  
1125  // rl.on('line', function (message) {  
1126  
1127  })  
1128  
1129  // rl.on('line', function (message) {  
1130  
1131  })  
1132  
1133  // rl.on('line', function (message) {  
1134  
1135  })  
1136  
1137  // rl.on('line', function (message) {  
1138  
1139  })  
1140  
1141  // rl.on('line', function (message) {  
1142  
1143  })  
1144  
1145  // rl.on('line', function (message) {  
1146  
1147  })  
1148  
1149  // rl.on('line', function (message) {  
1150  
1151  })  
1152  
1153  // rl.on('line', function (message) {  
1154  
1155  })  
1156  
1157  // rl.on('line', function (message) {  
1158  
1159  })  
1160  
1161  // rl.on('line', function (message) {  
1162  
1163  })  
1164  
1165  // rl.on('line', function (message) {  
1166  
1167  })  
1168  
1169  // rl.on('line', function (message) {  
1170  
1171  })  
1172  
1173  // rl.on('line', function (message) {  
1174  
1175  })  
1176  
1177  // rl.on('line', function (message) {  
1178  
1179  })  
1180  
1181  // rl.on('line', function (message) {  
1182  
1183  })  
1184  
1185  // rl.on('line', function (message) {  
1186  
1187  })  
1188  
1189  // rl.on('line', function (message) {  
1190  
1191  })  
1192  
1193  // rl.on('line', function (message) {  
1194  
1195  })  
1196  
1197  // rl.on('line', function (message) {  
1198  
1199  })  
1200  
1201  // rl.on('line', function (message) {  
1202  
1203  })  
1204  
1205  // rl.on('line', function (message) {  
1206  
1207  })  
1208  
1209  // rl.on('line', function (message) {  
1210  
1211  })  
1212  
1213  // rl.on('line', function (message) {  
1214  
1215  })  
1216  
1217  // rl.on('line', function (message) {  
1218  
1219  })  
1220  
1221  // rl.on('line', function (message) {  
1222  
1223  })  
1224  
1225  // rl.on('line', function (message) {  
1226  
1227  })  
1228  
1229  // rl.on('line', function (message) {  
1230  
1231  })  
1232  
1233  // rl.on('line', function (message) {  
1234  
1235  })  
1236  
1237  // rl.on('line', function (message) {  
1238  
1239  })  
1240  
1241  // rl.on('line', function (message) {  
1242  
1243  })  
1244  
1245  // rl.on('line', function (message) {  
1246  
1247  })  
1248  
1249  // rl.on('line', function (message) {  
1250  
1251  })  
1252  
1253  // rl.on('line', function (message) {  
1254  
1255  })  
1256  
1257  // rl.on('line', function (message) {  
1258  
1259  })  
1260  
1261  // rl.on('line', function (message) {  
1262  
1263  })  
1264  
1265  // rl.on('line', function (message) {  
1266  
1267  })  
1268  
1269  // rl.on('line', function (message) {  
1270
```


4. Use above chatbot module in web based chatting of websocket.

Q4.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title></title>
</head>

<body>
  <h1>WebSocket Chat Bot</h1>
  <div id="chat">
    <div id="messages"></div>
    <input type="text" id="inputMessage" placeholder="Type your message
here..." />
    <button onclick="sendMessage()">Send</button>
  </div>

  <script>
    const ws = new WebSocket('ws://localhost:4589');

    ws.onmessage = (event) => {
      displayMessage("Server: " + event.data);
    };

    function sendMessage() {
      const inputMessage = document.getElementById('inputMessage');
      const message = inputMessage.value;
      inputMessage.value = '';

      displayMessage('You: ' + message);
      ws.send(message);
    }

    function displayMessage(message) {
      const messagesDiv = document.getElementById('messages');
      const messageDiv = document.createElement('div');
      messageDiv.textContent = message;
      messagesDiv.appendChild(messageDiv);
    }
  </script>
</body>

</html>
```

Mainchat.js

```
const http = require('http');
const st = require('node-static');
const chatBot = require('../Q3/chatbot'); // Import chatbot.js module
const WebSocket = require('ws');

const file = new st.Server('./Q4.html');

const server = http.createServer((req, res) => {
  req.on('end', () => {
    file.serve(req, res);
  }).resume();
});

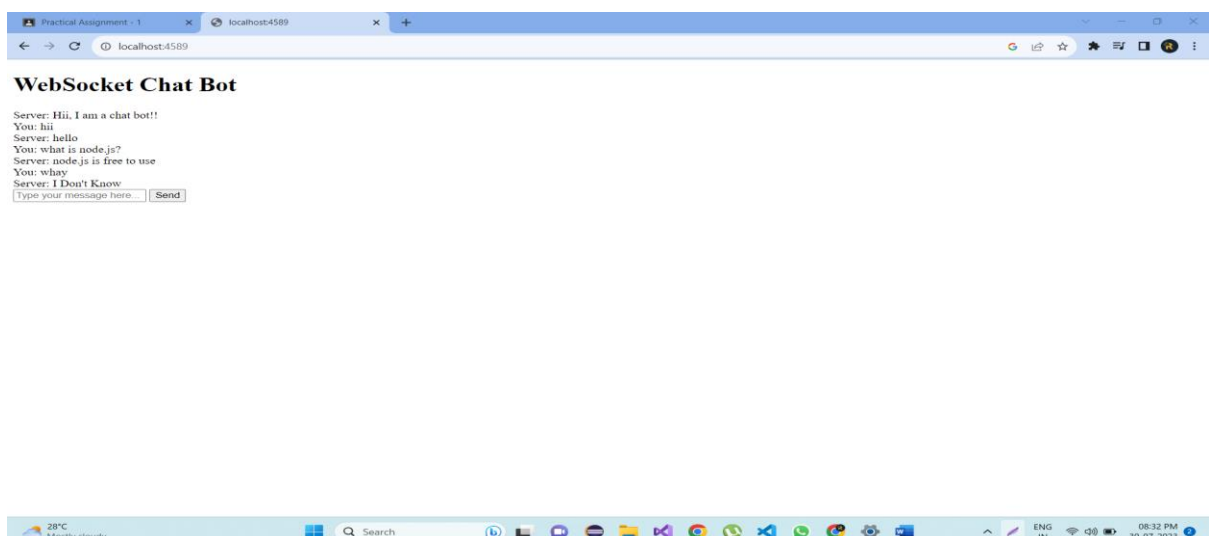
server.listen(4589, () => {
  console.log("Server listening on 4589");
});

const wss = new WebSocket.Server({ server: server });

wss.on('connection', (ws) => {
  ws.send("Hi, I am a chat bot!!");

  ws.on('message', (data) => {
    const message = data.toString();
    const reply = chatBot.chatreplay(message);
    ws.send(reply);
  });
});
```

Output:



5. Write a program to create a compressed zip file for a folder.

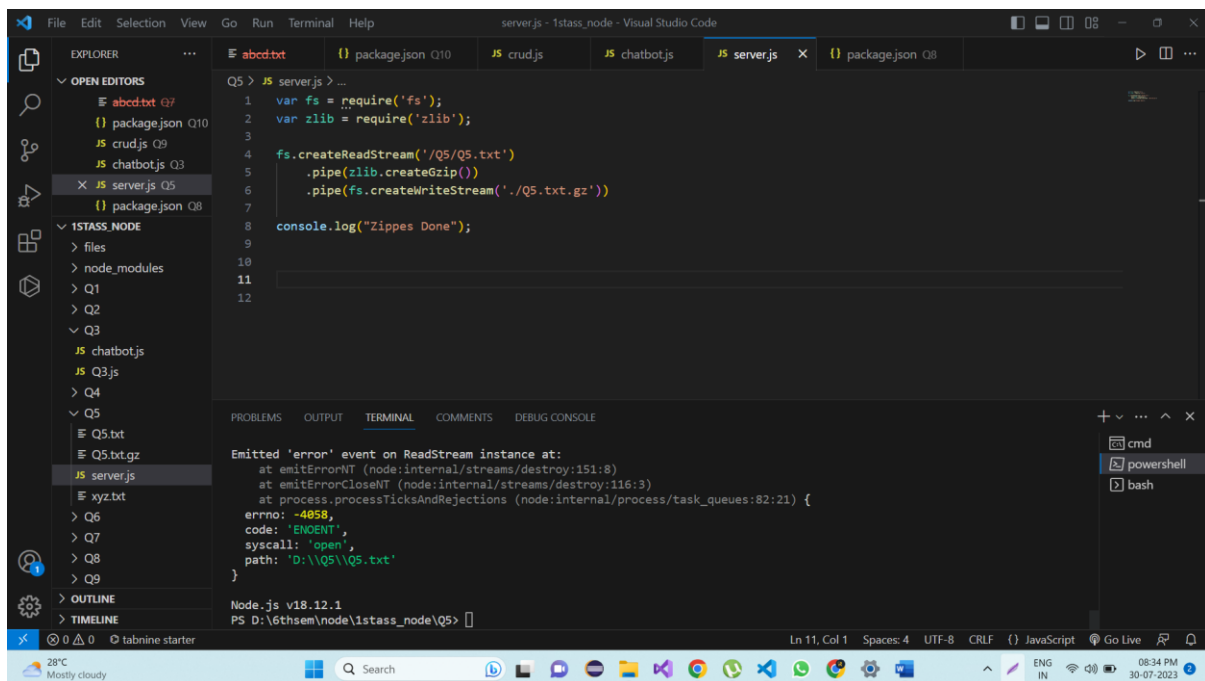
Server.js

```
var fs = require('fs');
var zlib = require('zlib');

fs.createReadStream('/Q5/Q5.txt')
  .pipe(zlib.createGzip())
  .pipe(fs.createWriteStream('./Q5.txt.gz'))

console.log("Zippes Done");
```

output:



6. Write a program to extract a zip file.

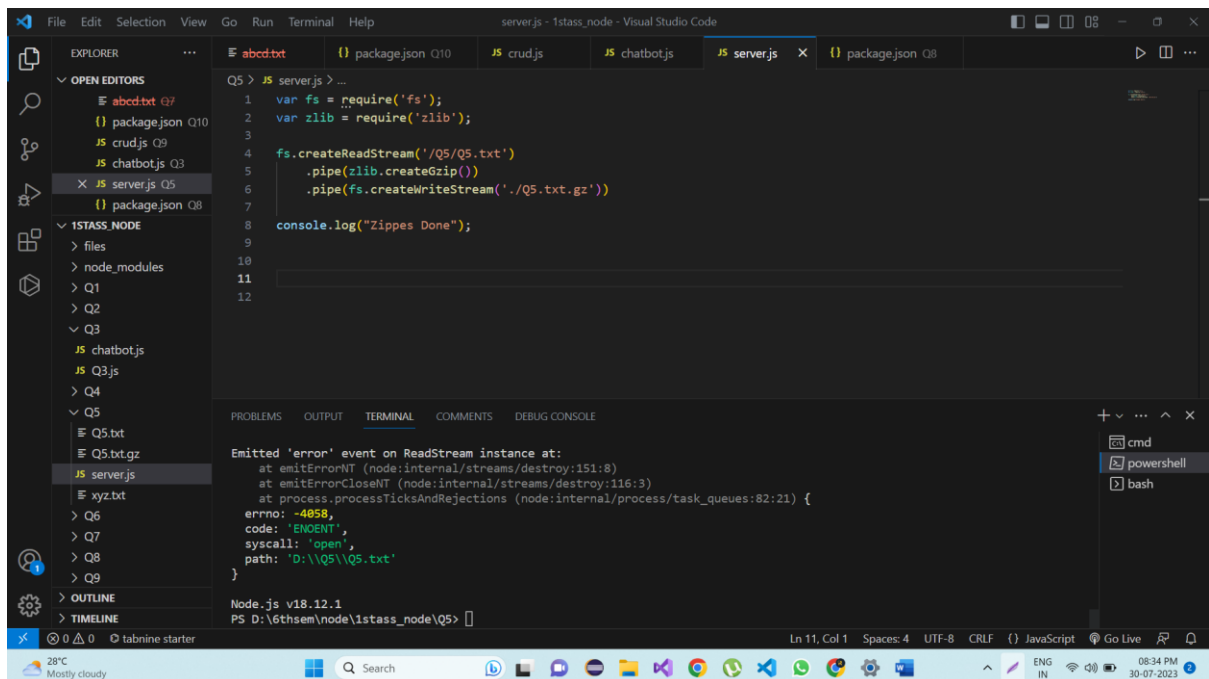
Decom.js

```
var fs = require('fs')
var zlib = require('zlib')

fs.createReadStream('../Q5/Q5.txt.gz')
  .pipe(zlib.createGunzip())
  .pipe(fs.createWriteStream('../Q5/xyz.txt', 'utf-8'))

console.log("dcompress done")
```

output:



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project structure with folders like 'files', 'node_modules', and 'Q1' through 'Q9'. The 'serverjs' file is selected. The main editor shows the following JavaScript code:

```
1 var fs = require('fs');
2 var zlib = require('zlib');
3
4 fs.createReadStream('/Q5/Q5.txt')
5   .pipe(zlib.createGzip())
6   .pipe(fs.createWriteStream('./Q5.txt.gz'))
7
8 console.log("Zippes Done");
9
10
11
12
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
Emitted 'error' event on ReadStream instance at:
  at emitErrorNT (node:internal/streams/destroy:151:8)
  at emitErrorCloseNT (node:internal/streams/destroy:116:3)
  at process.processTicksAndRejections (node:internal/process/task_queues:82:21) {
  errno: -4058,
  code: 'ENOENT',
  syscall: 'open',
  path: 'D:\\Q5\\Q5.txt'
}
```

The status bar at the bottom indicates the file is 'Node.js v18.12.1' and the terminal is running 'PS D:\6thsem\node\1stass_node\Q5>'.

7. Write a program to promisify fs.unlink function and call it.

Promiseunlink.js

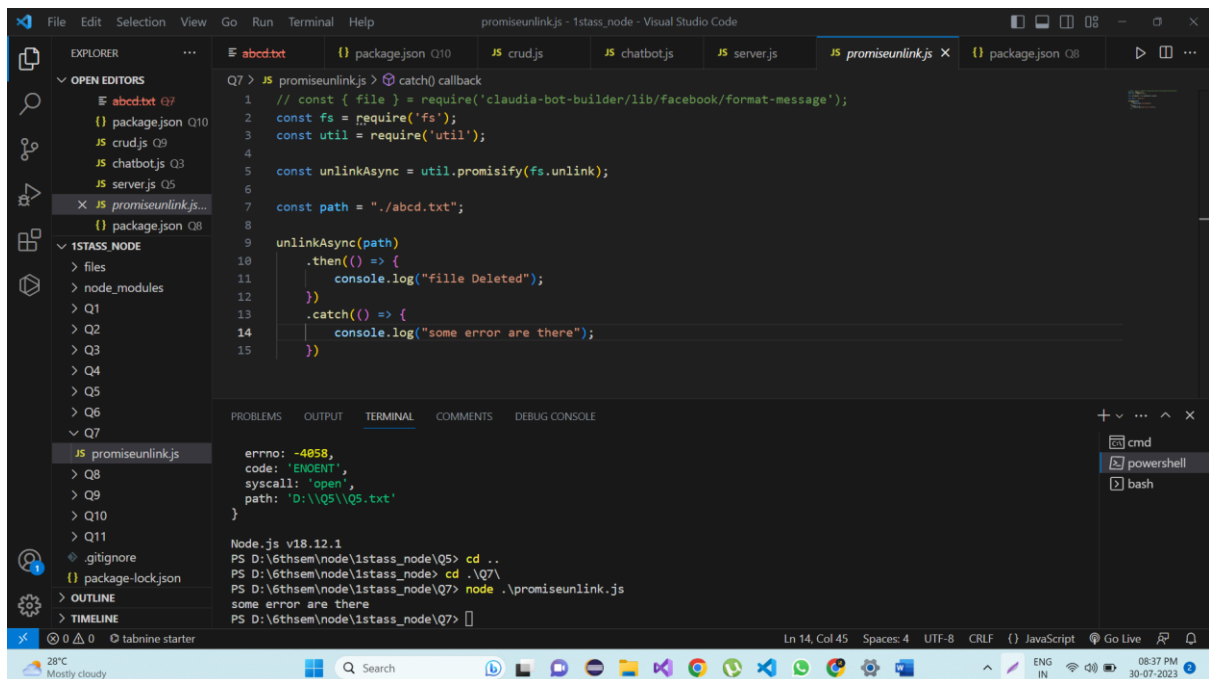
```
// const { file } = require('claudia-bot-builder/lib/facebook/format-
message');
const fs = require('fs');
const util = require('util');

const unlinkAsync = util.promisify(fs.unlink);

const path = "./abcd.txt";

unlinkAsync(path)
  .then(() => {
    console.log("file Deleted");
  })
  .catch(() => {
    console.log("some error are there");
  })
})
```

Output:



The screenshot shows the Visual Studio Code editor with a file named 'promiseunlink.js' open. The code in the file is as follows:

```
1 // const { file } = require('claudia-bot-builder/lib/facebook/format-message');
2 const fs = require('fs');
3 const util = require('util');
4
5 const unlinkAsync = util.promisify(fs.unlink);
6
7 const path = './abcd.txt';
8
9 unlinkAsync(path)
10 .then(() => {
11   console.log("file Deleted");
12 })
13 .catch(() => {
14   console.log("some error are there");
15 })
```

The terminal output shows an error: 'errno: -4058, code: 'ENOENT', syscall: 'open', path: 'D:\Q5\Q5.txt''. Below the error, the terminal shows the command 'node .\promiseunlink.js' being executed, which results in the message 'some error are there'.

8. Fetch data of google page using node-fetch using async-await model.

Q8.js

```
const http = require('http');
const server = http.createServer((req, res) => {
  async function fetchGooglePage() {
    try {
      const fetch = await import('node-fetch');
      const response = await fetch.default('https://www.google.com');

      if (!response.ok) {
        throw new Error('Network response was not ok');
      }

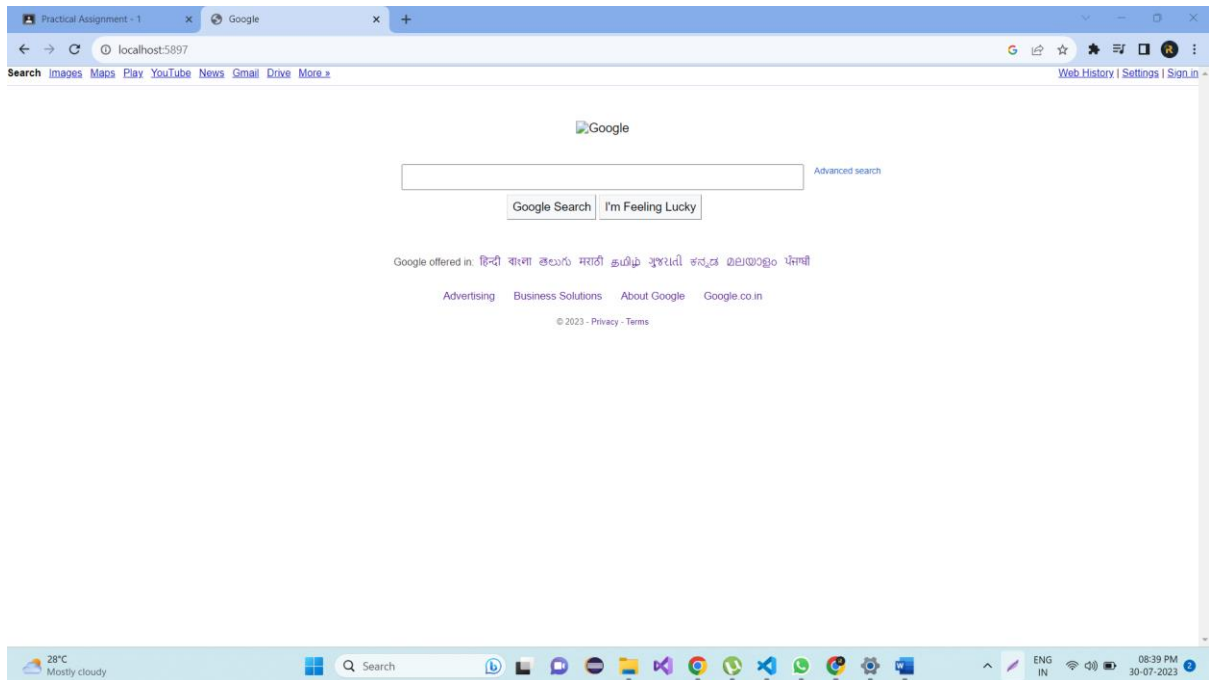
      const data = await response.text();
      // console.log(data);
      res.end(data);
    } catch (error) {
      console.error('Error fetching data:', error.message);
    }
  }

  fetchGooglePage();
})

server.listen(5897, () => {
  console.log("Listing on 5897");
});
```

})

Output:



9. Write a program that connect Mysql database, Insert a record in employee table and display all records in employee table using promise based approach.

Crud.js

```
const mysql = require('nodejs-mysql').default;

const conn = ({
  host: "localhost",
  user: "root",
  password: "",
  database: "Node_ass"
});

const db = mysql.getInstance(conn);

db.connect()
  .then(() => {
    console.log(`Connected!!`)

    var sql = "INSERT INTO employeetb (empid,empname,joinDate) VALUES (201,'abc','25-06-2022')";
    console.log("Record Inserted!!");
    return db.exec(sql);
  })
```

```

.then((display) => {
    // var sqlDisplay = "SELECT * FROM employeeetb";
    // console.log(display);
    return db.exec("SELECT * FROM employeeetb");
})

.then((result) => {
    console.log('Employee Name \t Date of Join');
    for (var i in result) {
        console.log(result[i].empname + " \t\t " + result[i].joindate);
    }
})

.catch((err) => {
    console.log("Error: " + err);
    process.exit(0);
})

```

Output:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists the project files. The main editor window shows the JavaScript code. The TERMINAL panel at the bottom displays the output of running the script:

```

abc undefined
PS D:\6thsem\node\1stass_node\Q9> node .\crud.js
Connected!!
Record Inserted!!
Employee Name Date of Join
abcd 21-aug-2023
abcd 21-aug-2023
abcd 21-aug-2023
abcd 21-aug-2023
abc 25-06-2022
abc 25-06-2022

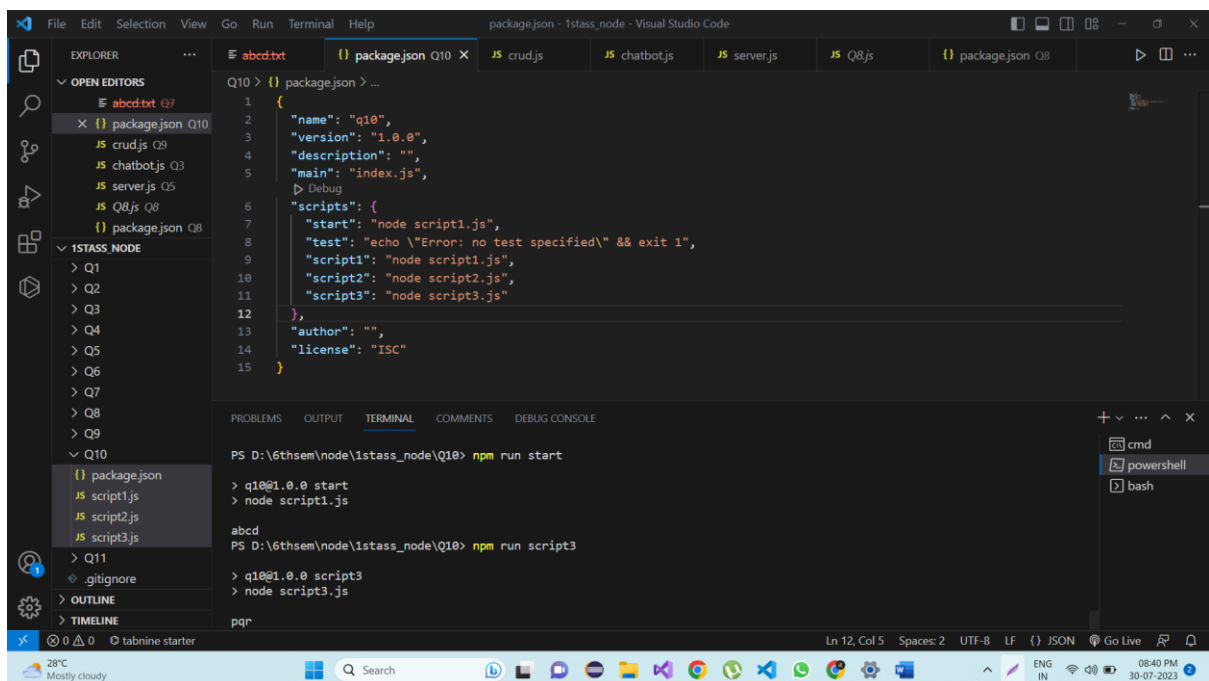
```

10. Set a server script, a test script and 3 user defined scripts in package.json file in your nodejs application.

Package.json

```
{
  "name": "q10",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node script1.js",
    "test": "echo \"Error: no test specified\" && exit 1",
    "script1": "node script1.js",
    "script2": "node script2.js",
    "script3": "node script3.js"
  },
  "author": "",
  "license": "ISC"
}
```

Output:



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure with files like `package.json`, `script1.js`, `script2.js`, and `script3.js`. The main editor area shows the `package.json` file with the following content:

```
{
  "name": "q10",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node script1.js",
    "test": "echo \"Error: no test specified\" && exit 1",
    "script1": "node script1.js",
    "script2": "node script2.js",
    "script3": "node script3.js"
  },
  "author": "",
  "license": "ISC"
}
```

The bottom panel shows the Terminal with the following commands and output:

```
PS D:\6thsem\node\1stass_node\Q10> npm run start
> q10@1.0.0 start
> node script1.js

abcd
PS D:\6thsem\node\1stass_node\Q10> npm run script3
> q10@1.0.0 script3
> node script3.js

pqr
```


11. Develop an application to show live cricket score.

Crick.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">
  <link rel="stylesheet" type="text/css" href="./css/bootstrap.min.css">
  <script src="./js/bootstrap.bundle.js" defer></script>
  <title>Document</title>
</head>

<body>
  <div class="container my-4">
    <h2 class="my-4">Cricket Score</h2>
    <div class="d-flex justify-content-around">
      <div class="row">

        <div class="card text mb-3">
          <div class="card-header" id="card-header1"></div>
          <div class="card-body">
            <h5 class="card-title" id="card-title1"></h5>
            <p class="card-text">
              <div id="batting-team1"></div>
              <div id="match-score1"></div>
            </p>
          </div>
        </div>
        <div class="card text mb-3">
          <div class="card-header" id="card-header2"></div>
          <div class="card-body">
            <h5 class="card-title" id="card-title2"></h5>
            <p class="card-text">
              <div id="batting-team2"></div>
              <div id="match-score2"></div>
            </p>
          </div>
        </div>
        <div class="card mb-3">
          <div class="card-header" id="card-header3"></div>
          <div class="card-body">
            <h5 class="card-title" id="card-title3"></h5>
            <p class="card-text">
```

```

        <div id="batting-team3"></div>
        <div id="match-score3"></div>

    </p>
    </div>
  </div>
</div>
</body>
<script>
  const ws = new WebSocket("ws://localhost:3300");
  ws.addEventListener("message", (msg) => {
    let res = JSON.parse(msg.data);
    console.log(res);
    let match = res[1];
    document.getElementById("card-header1").innerText = match.name;
    document.getElementById("card-title1").innerText = match.status;
    document.getElementById("batting-team1").innerText =
match.score[0].inning;
    document.getElementById("match-score1").innerText = match.score[0].r +
 '/' + res[0].score[1].w + ' Over : ' + res[0].score[1].o;

    let match2 = res[2];
    document.getElementById("card-header2").innerText = match2.name;
    document.getElementById("card-title2").innerText = match2.status;
    document.getElementById("batting-team2").innerText =
match2.score[0].inning;
    document.getElementById("match-score2").innerText = match2.score[0].r
+ '/' + res[0].score[1].w + ' Over : ' + res[0].score[1].o;

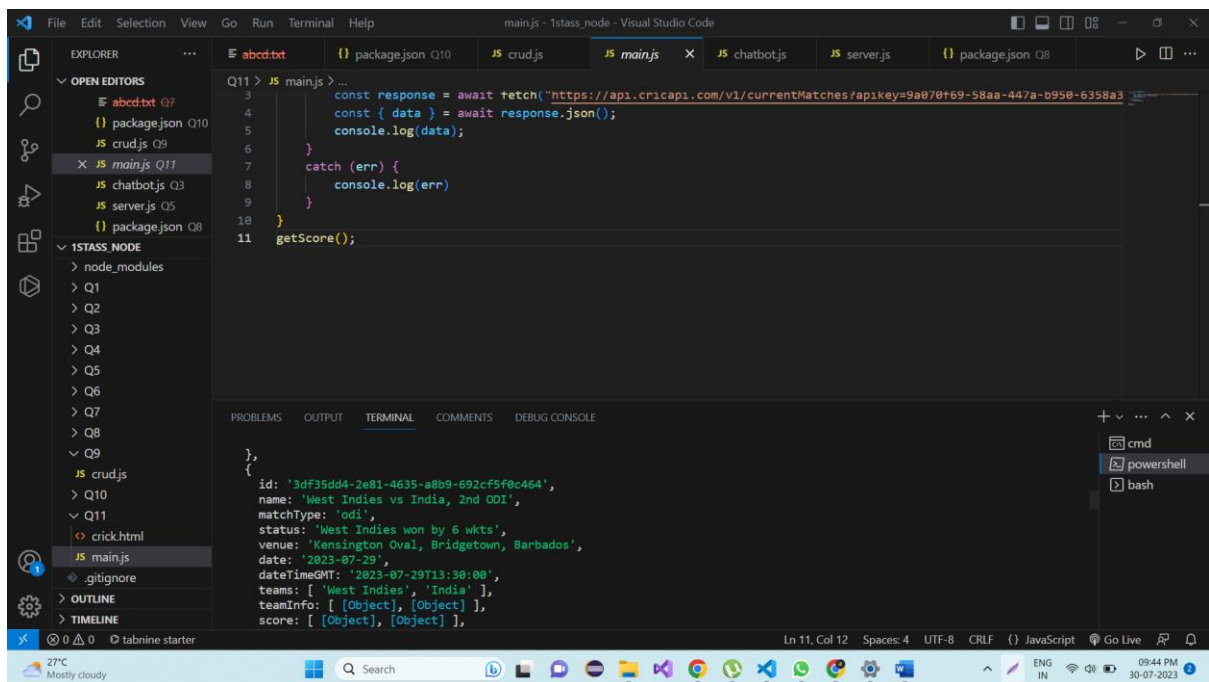
    let match3 = res[3];
    document.getElementById("card-header3").innerText = match3.name;
    document.getElementById("card-title3").innerText = match3.status;
    document.getElementById("batting-team3").innerText =
match3.score[0].inning;
    document.getElementById("match-score3").innerText = match3.score[0].r
+ '/' + res[0].score[1].w + ' Over : ' + res[0].score[1].o;
  })
</script>
</html>

```

Main.js

```
async function getScore() {
  try {
    const response = await
fetch("https://api.cricapi.com/v1/currentMatches?apikey=9a070f69-58aa-447a-
b950-6358a39bf75e&offset=0");
    const { data } = await response.json();
    console.log(data);
  }
  catch (err) {
    console.log(err)
  }
}
getScore();
```

output:



```
Q11> JS main.js > ...
3      const response = await fetch("https://api.cricapi.com/v1/currentMatches?apikey=9a070f69-58aa-447a-b950-6358a3
4      const { data } = await response.json();
5      console.log(data);
6    }
7    catch (err) {
8      console.log(err)
9    }
10  }
11  getScore();
```

```
},
{
  id: '3df35dd4-2e81-4635-a8b9-692cf5f8c464',
  name: 'West Indies vs India, 2nd ODI',
  matchType: 'odi',
  status: 'West Indies won by 6 wkts',
  venue: 'Kensington Oval, Bridgetown, Barbados',
  date: '2023-07-29',
  dateTimeGMT: '2023-07-29T13:30:00',
  teams: [ 'West Indies', 'India' ],
  teamInfo: [ [Object], [Object] ],
  score: [ [Object], [Object] ],
```