

Dictionary Project Report

Contents

Screenshots 2

Source codes..... 5

Figures

FIGURE 1: EMPTY INPUT VALIDATION..... 2

FIGURE 2 : DICTIONARY WORD SEARCH 3

FIGURE 3:SAMPLE TERMINAL CONSOLE OUTPUT 4

FIGURE 4: DICTIONARY.JS 5

FIGURE 5: DICT.HTML..... 6

FIGURE 6: DICT.JS 7

FIGURE 7: WORD.JS 8

FIGURE 8:CSS [DISCT.CSS]..... 9

FIGURE 9:PROJECT STRUCTURE AND PACKAGE.JSON 10

Dictionary Project Report

Screenshots

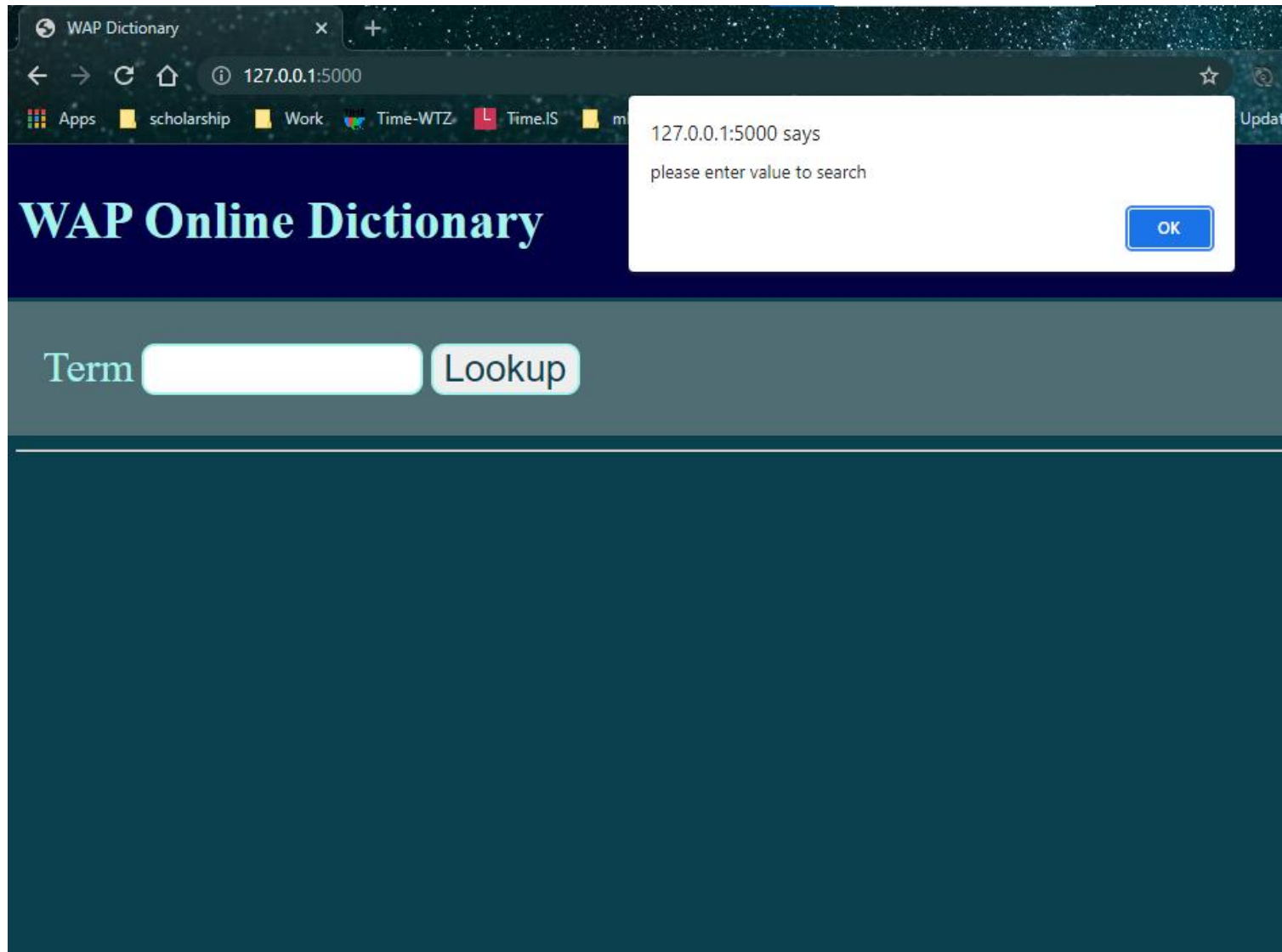


Figure 1: Empty input validation

Dictionary Project Report

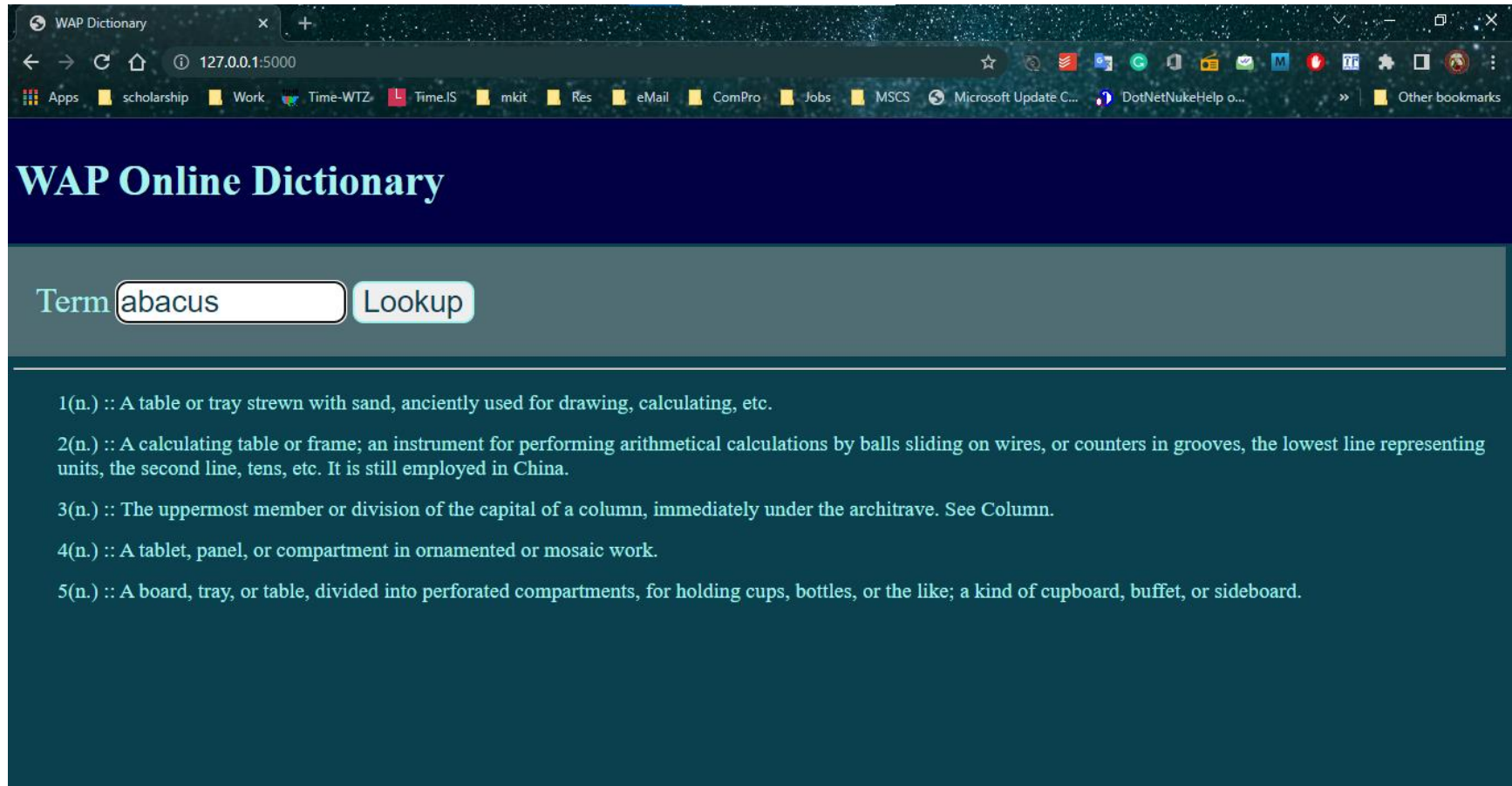
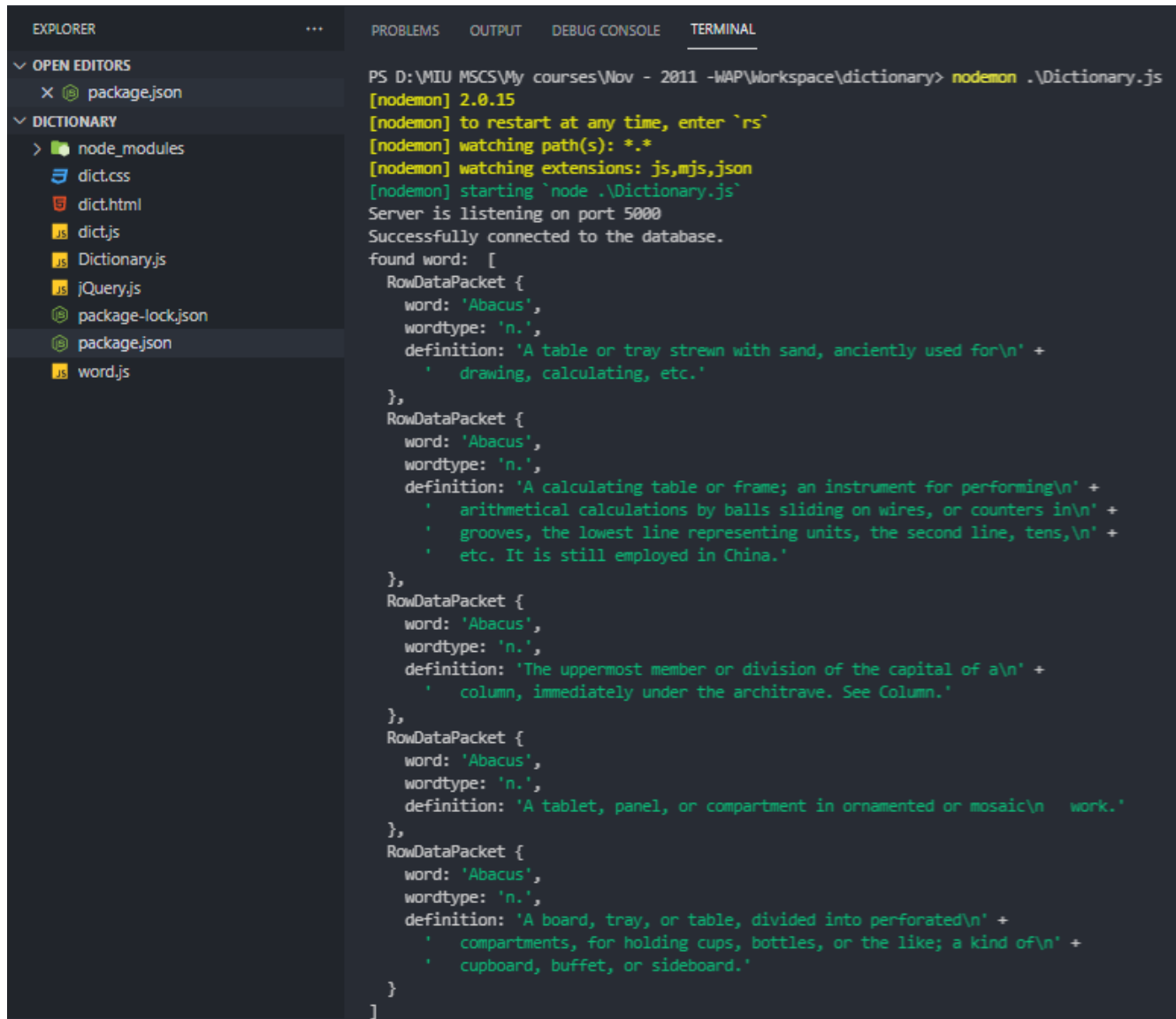


Figure 2 : Dictionary word search

Dictionary Project Report



```
EXPLORER    ...    PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

v OPEN EDITORS
  x package.json
v DICTIONARY
  > node_modules
    dict.css
    dict.html
    dict.js
    Dictionary.js
    jQuery.js
    package-lock.json
    package.json
    word.js

PS D:\MIU MSCS\My courses\Nov - 2011 -WAP\Workspace\dictionary> nodemon .\Dictionary.js
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node .\Dictionary.js`
Server is listening on port 5000
Successfully connected to the database.
found word: [
  RowDataPacket {
    word: 'Abacus',
    wordtype: 'n.',
    definition: 'A table or tray strewn with sand, anciently used for\n' +
      '  drawing, calculating, etc.'
  },
  RowDataPacket {
    word: 'Abacus',
    wordtype: 'n.',
    definition: 'A calculating table or frame; an instrument for performing\n' +
      '  arithmetical calculations by balls sliding on wires, or counters in\n' +
      '  grooves, the lowest line representing units, the second line, tens,\n' +
      '  etc. It is still employed in China.'
  },
  RowDataPacket {
    word: 'Abacus',
    wordtype: 'n.',
    definition: 'The uppermost member or division of the capital of a\n' +
      '  column, immediately under the architrave. See Column.'
  },
  RowDataPacket {
    word: 'Abacus',
    wordtype: 'n.',
    definition: 'A tablet, panel, or compartment in ornamented or mosaic\n  work.'
  },
  RowDataPacket {
    word: 'Abacus',
    wordtype: 'n.',
    definition: 'A board, tray, or table, divided into perforated\n' +
      '  compartments, for holding cups, bottles, or the like; a kind of\n' +
      '  cupboard, buffet, or sideboard.'
  }
]
```

Figure 3: Sample Terminal console output

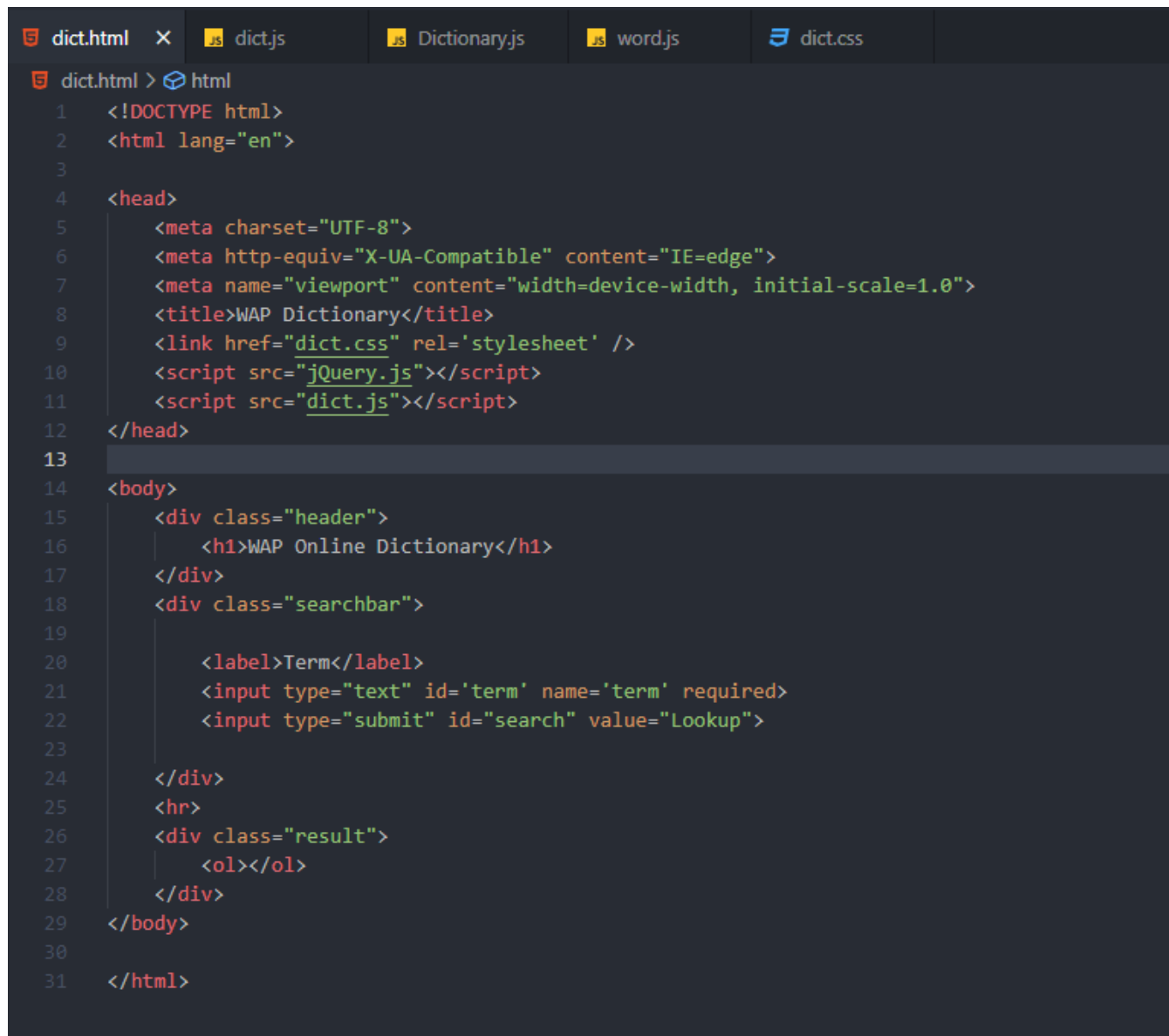
Dictionary Project Report

Source codes

```
Dictionary.js > app.post('/search') callback
1  const express = require('express');
2
3  const app = express();
4  app.use(express.urlencoded({
5    extended: false
6  }));
7
8  app.use(express.static('.'));
9
10 app.get('/', (req, res) => {
11   res.sendFile(__dirname + '\\dict.html');
12 });
13 app.post('/search', (req, res) => {
14   let searchKey = req.body.key;
15
16   var keyDefinitions = require("./word.js");
17   keyDefinitions.search(searchKey, res);
18 });
19
20
21 let port = 5000;
22 app.listen(port, () => console.log(`Server is listening on port ${port} `));
```

Figure 4: Dictionary.js

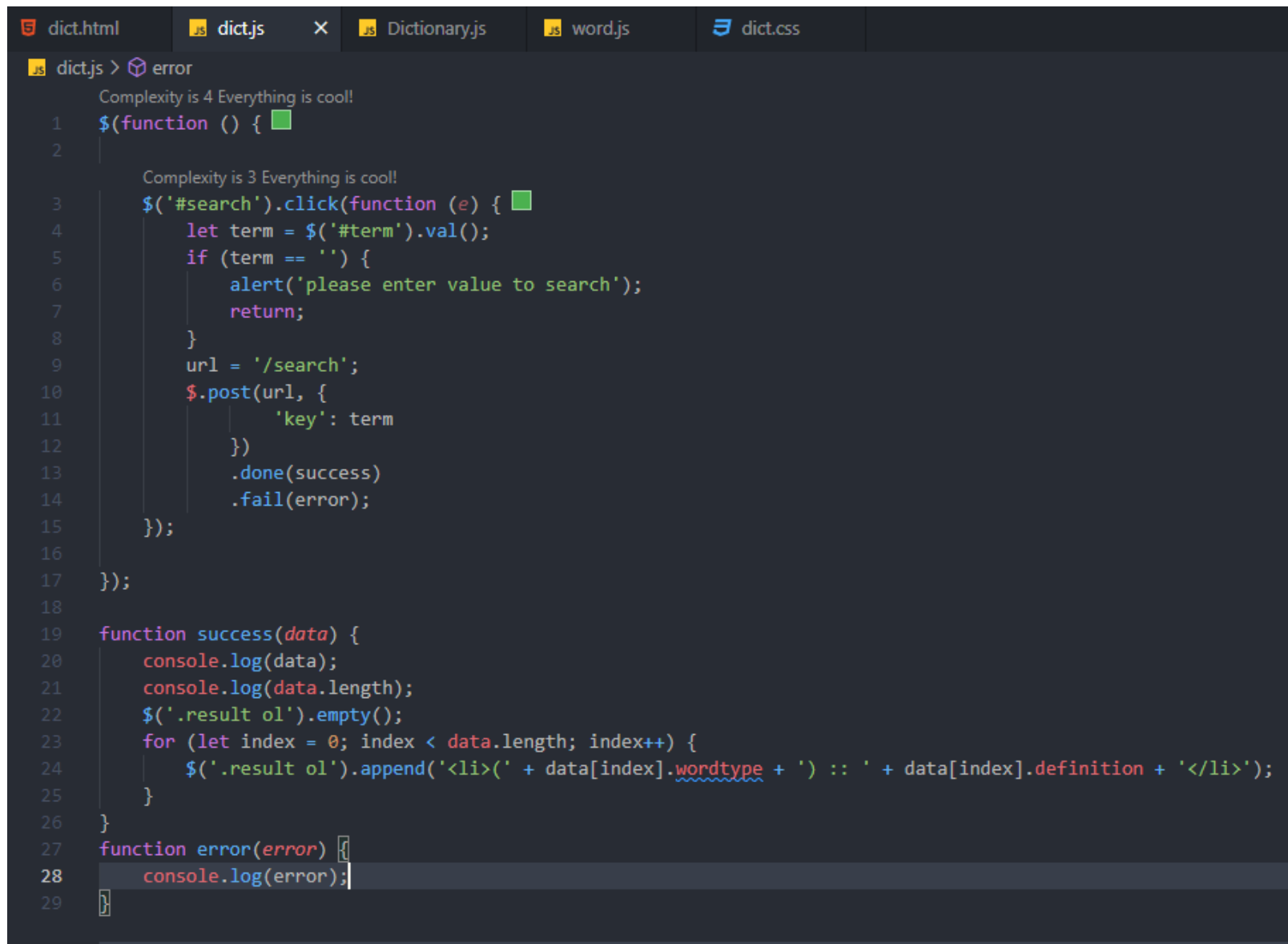
Dictionary Project Report



```
dict.html x dict.js Dictionary.js word.js dict.css
dict.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>WAP Dictionary</title>
9     <link href="dict.css" rel='stylesheet' />
10    <script src="jQuery.js"></script>
11    <script src="dict.js"></script>
12 </head>
13
14 <body>
15     <div class="header">
16         <h1>WAP Online Dictionary</h1>
17     </div>
18     <div class="searchbar">
19
20         <label>Term</label>
21         <input type="text" id='term' name='term' required>
22         <input type="submit" id="search" value="Lookup">
23
24     </div>
25     <hr>
26     <div class="result">
27         <ol></ol>
28     </div>
29 </body>
30
31 </html>
```

Figure 5: dict.html

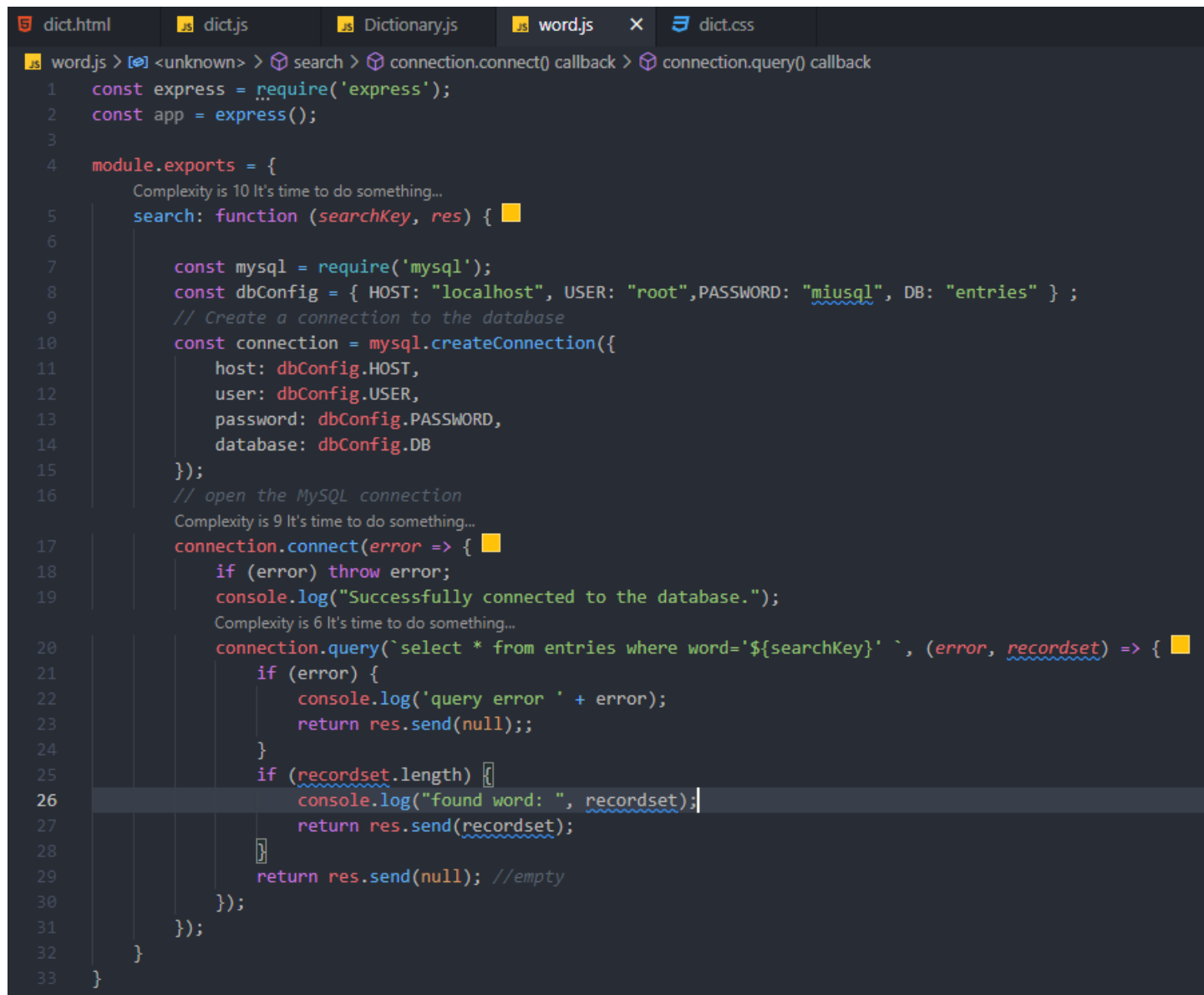
Dictionary Project Report



```
dict.js > error
Complexity is 4 Everything is cool!
1 $(function () {
2
3     Complexity is 3 Everything is cool!
4     $('#search').click(function (e) {
5         let term = $('#term').val();
6         if (term == '') {
7             alert('please enter value to search');
8             return;
9         }
10        url = '/search';
11        $.post(url, {
12            'key': term
13        })
14        .done(success)
15        .fail(error);
16    });
17 });
18
19 function success(data) {
20     console.log(data);
21     console.log(data.length);
22     $('#result ol').empty();
23     for (let index = 0; index < data.length; index++) {
24         $('#result ol').append('<li>(' + data[index].wordtype + ') :: ' + data[index].definition + '</li>');
25     }
26 }
27 function error(error) {
28     console.log(error);
29 }
```

Figure 6: dict.js

Dictionary Project Report



```
dict.html dict.js Dictionary.js word.js dict.css
word.js > <unknown> > search > connection.connect() callback > connection.query() callback
1  const express = require('express');
2  const app = express();
3
4  module.exports = {
    Complexity is 10 It's time to do something...
5    search: function (searchKey, res) {
6
7      const mysql = require('mysql');
8      const dbConfig = { HOST: "localhost", USER: "root",PASSWORD: "miusql", DB: "entries" } ;
9      // Create a connection to the database
10     const connection = mysql.createConnection({
11       host: dbConfig.HOST,
12       user: dbConfig.USER,
13       password: dbConfig.PASSWORD,
14       database: dbConfig.DB
15     });
16     // open the MySQL connection
17     Complexity is 9 It's time to do something...
18     connection.connect(error => {
19       if (error) throw error;
20       console.log("Successfully connected to the database.");
21       Complexity is 6 It's time to do something...
22       connection.query(`select * from entries where word='${searchKey}' `, (error, recordset) => {
23         if (error) {
24           console.log('query error ' + error);
25           return res.send(null);
26         }
27         if (recordset.length) {
28           console.log("found word: ", recordset);
29           return res.send(recordset);
30         }
31         return res.send(null); //empty
32       });
33     });
34   }
35 }
```

Figure 7: word.js

Dictionary Project Report

```
dict.css dict.css\ .searchbar input
body {
  color: #a7f4ee;
  background-color: #0b404e;
  font-size: larger;
}
.header {
  color: #a7f4ee;
  background-color: #020044;
  padding: 10px 10px;
  margin: -10px -10px 3px -8px;
}
.searchbar {
  padding: 30px;
  margin: auto;
  margin-left: -10px;
  font-size: larger;
  background-color: #506d74;
}
.searchbar label {
  font-size: xx-large;
}
.searchbar input {
  width: 110px;
  height: auto;
  font-size: larger;
  color: #0b404e;
  border: #a7f4ee solid 2px;
  border-radius: 10px;
}
.searchbar input[type='text'] {
  width: 200px;
}
ol {
  counter-reset: item;
  list-style-type: none;
}
li {
  display: block;
  padding-bottom: 15px;
}
li:before {
  content: counter(item) "";
  counter-increment: item
}
```

Figure 8: CSS [dict.css]

Dictionary Project Report

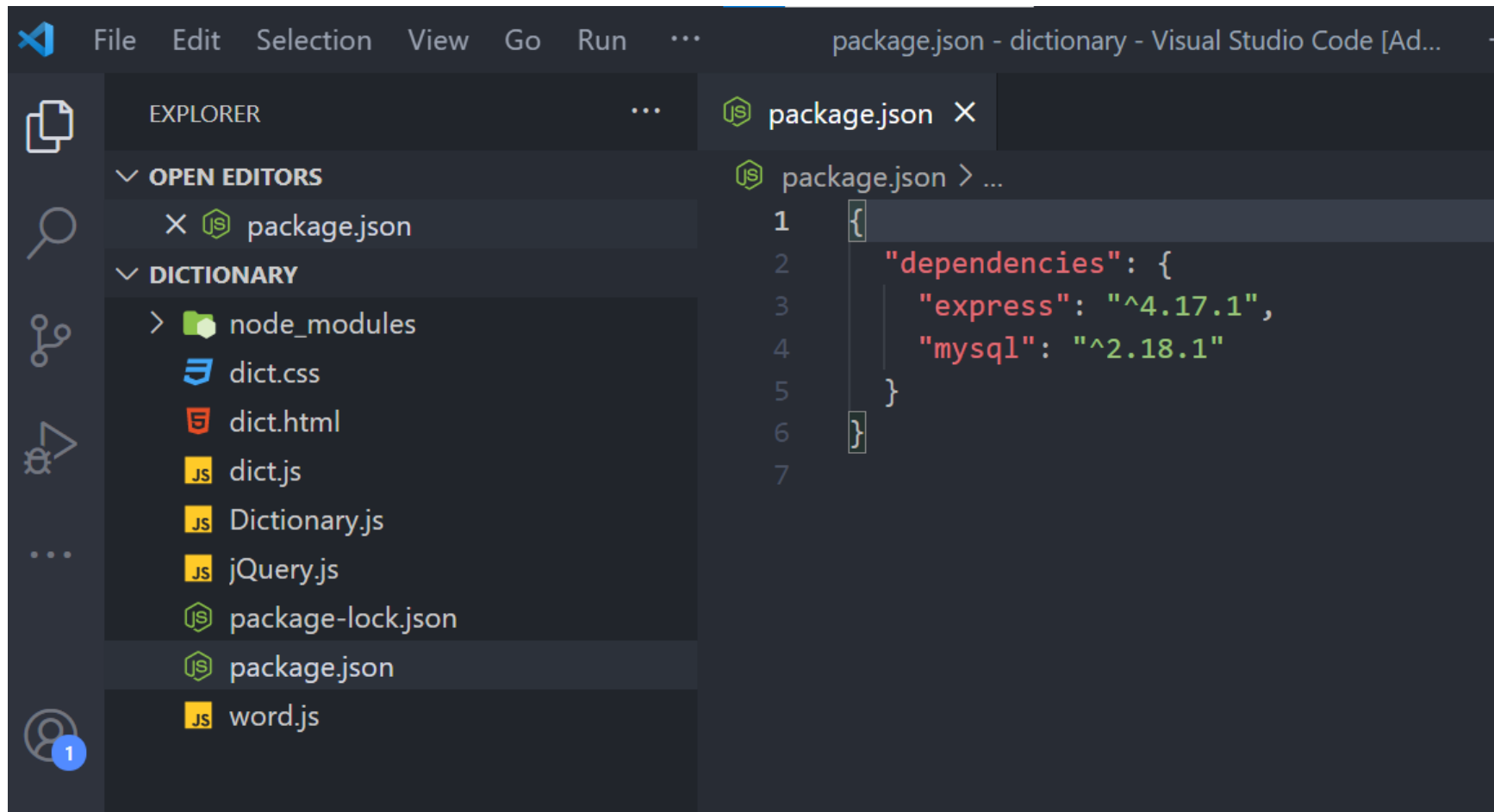


Figure 9: Project structure and package.json