Javascript-Socket.io

Building Portfolio page with Parallax effect, Using http-server, Saving codebase using GIT

Objectives of the Session:

- Usage Express and Socket.io

Prerequisites:

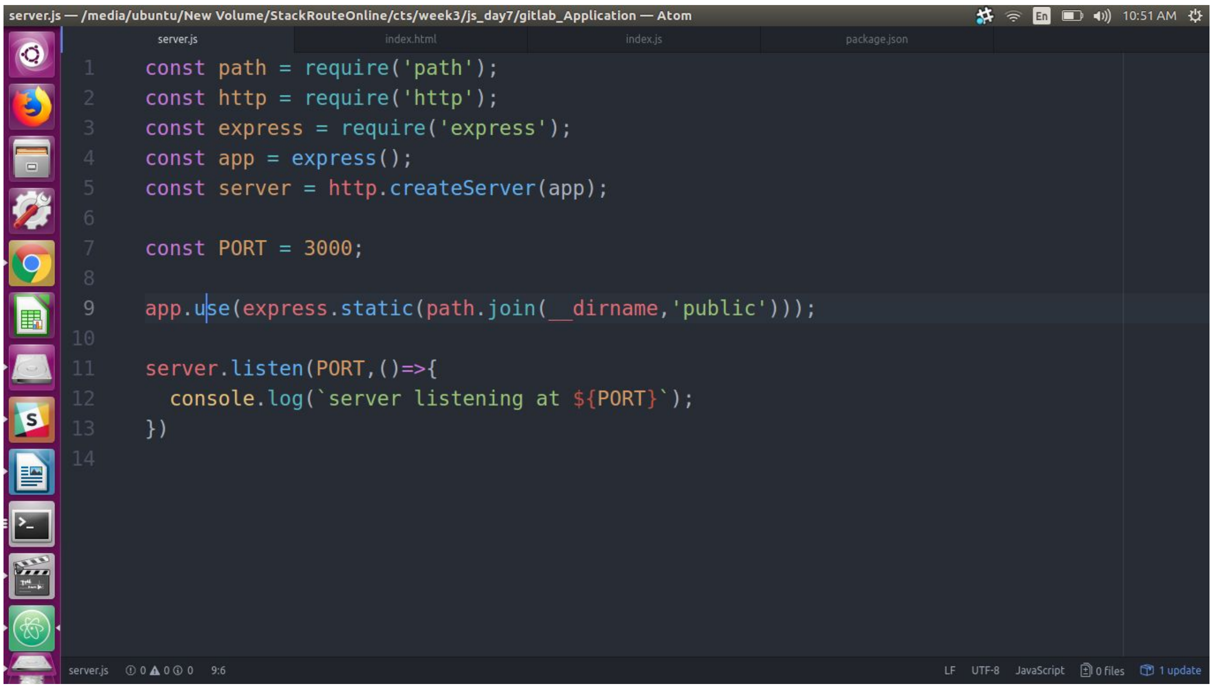
Basic understanding of html, css, bootstrap and JavaScript.

Learning:

In this session we will learn how to add chat in gitlab.

1) Create a public folder
2) Shift all the existing code in this folder.
3) Change the access key
4) Create a package.json
5) Add express using npm install express --save

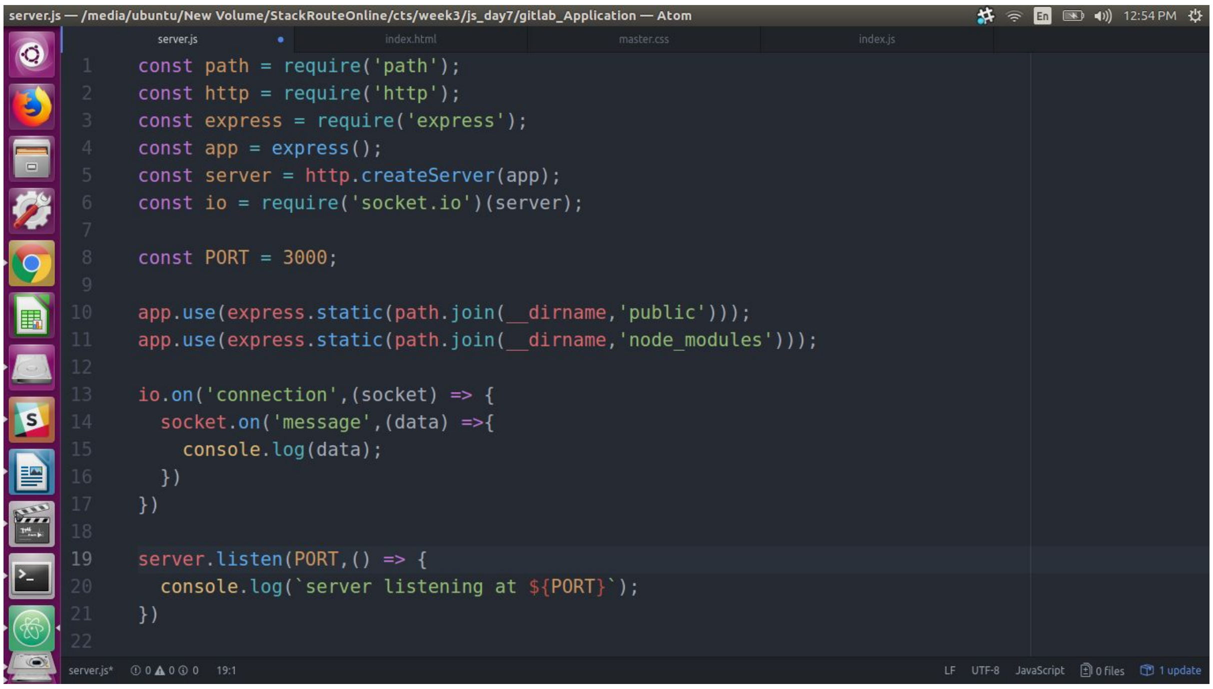
6) Create the server.js with following code



server.js — /media/ubuntu/New Volume/StackRouteOnline/cts/week3/js_day7/gitlab_Application — Atom

```
const path = require('path');
const http = require('http');
const express = require('express');
const app = express();
const server = http.createServer(app);

const PORT = 3000;

app.use(express.static(path.join(__dirname,'public')));

server.listen(PORT,()=>{
  console.log(`server listening at ${PORT}`);
})
```

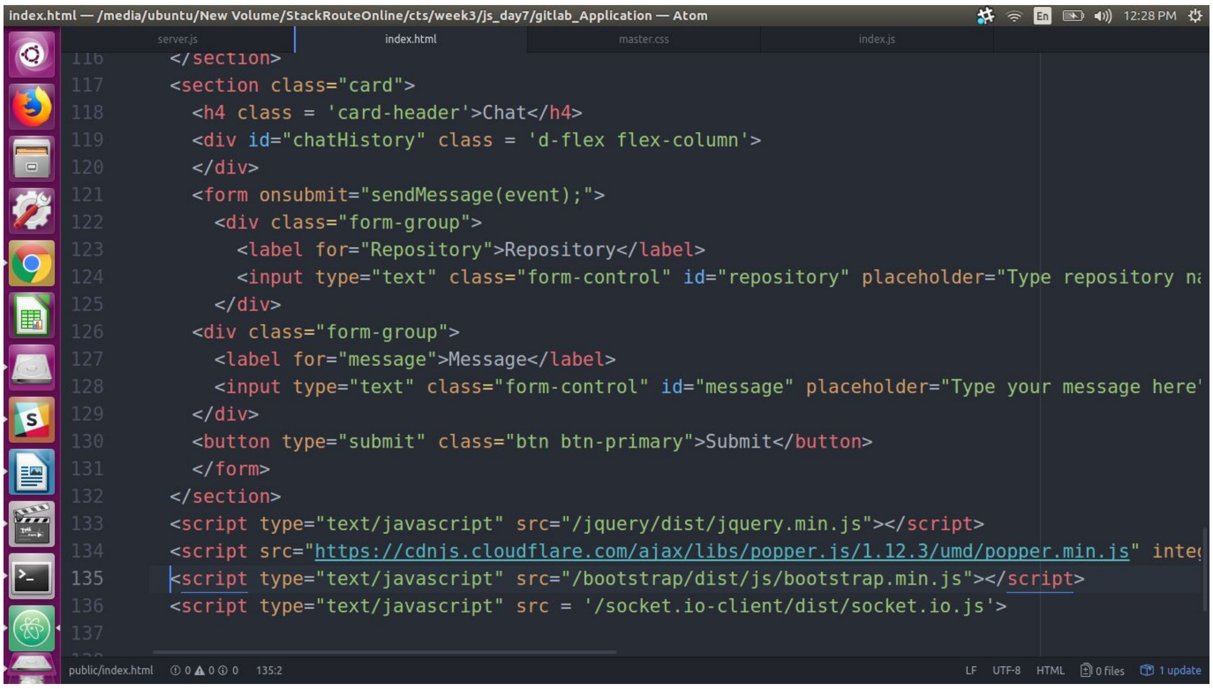
7) Add the start script.
8) Run the application using npm start.
9) Install bootstrap and use it.
10) Add the socket code.



server.js — /media/ubuntu/New Volume/StackRouteOnline/cts/week3/js_day7/gitlab_Application — Atom

```
const path = require('path');
const http = require('http');
const express = require('express');
const app = express();
const server = http.createServer(app);
const io = require('socket.io')(server);

const PORT = 3000;

app.use(express.static(path.join(__dirname,'public')));
app.use(express.static(path.join(__dirname,'node_modules')));

io.on('connection',(socket) => {
  socket.on('message',(data) =>{
    console.log(data);
  })
})

server.listen(PORT,() => {
  console.log(`server listening at ${PORT}`);
})
```

11) Add Chat UI.


index.html — /media/ubuntu/New Volume/StackRouteOnline/cts/week3/js_day7/gitlab_Application — Atom
12:28 PM
server.js
index.html
master.css
index.js
public/index.html
135:2
LF
UTF-8
HTML
0 files
1 update

```html
  </section>
  <section class="card">
    <h4 class = 'card-header'>Chat</h4>
    <div id="chatHistory" class = 'd-flex flex-column'>
    </div>
    <form onsubmit="sendMessage(event);">
      <div class="form-group">
        <label for="Repository">Repository</label>
        <input type="text" class="form-control" id="repository" placeholder="Type repository na
      </div>
      <div class="form-group">
        <label for="message">Message</label>
        <input type="text" class="form-control" id="message" placeholder="Type your message here'
      </div>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </section>
  <script type="text/javascript" src="/jquery/dist/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js" inte
  <script type="text/javascript" src="/bootstrap/dist/js/bootstrap.min.js"></script>
  <script type="text/javascript" src = '/socket.io-client/dist/socket.io.js'>
```

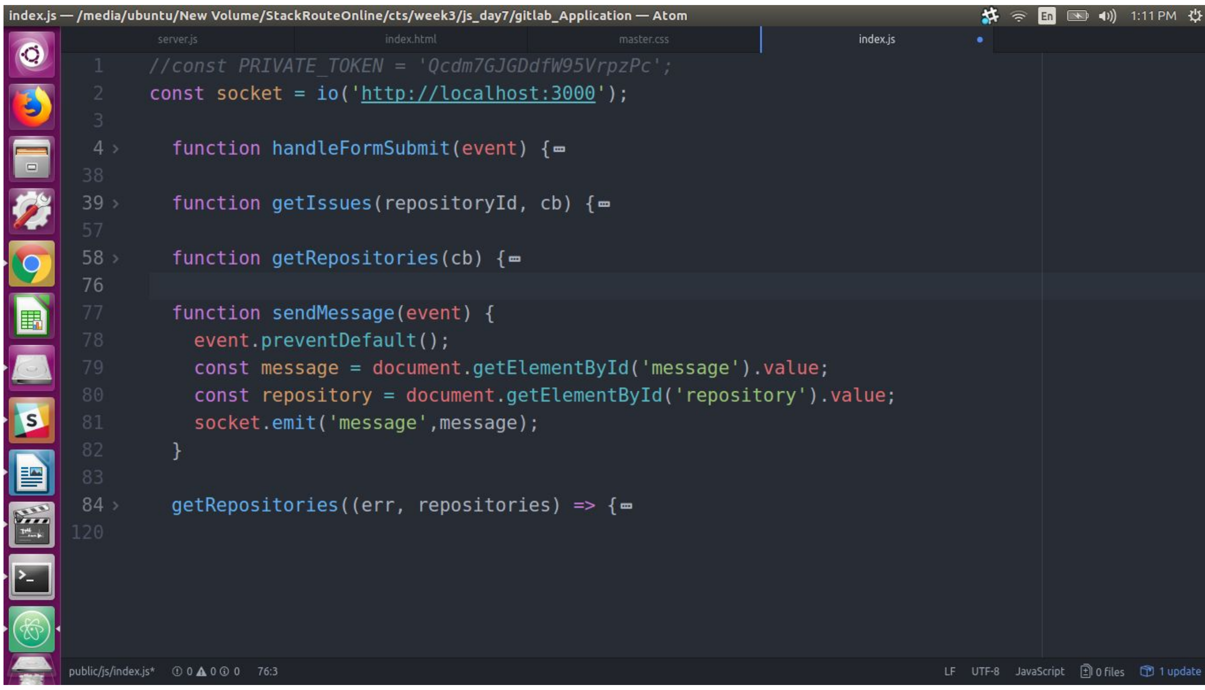
12) Add socket lib.<script type="text/javascript" src = '/socket.io-client/dist/socket.io.js'>

13) </script>

14) Add the client side code

a) Add const socket = io('http://localhost:3000'); at the top

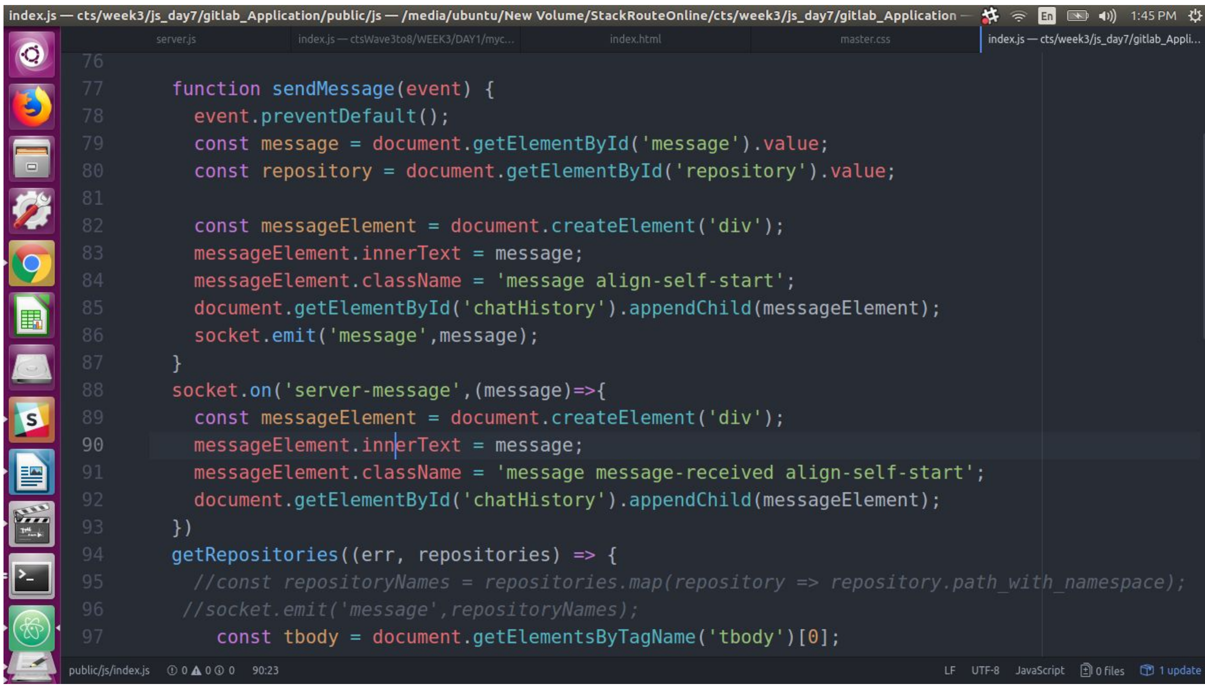
b) Add sendMessage Method with following code.



```javascript
//const PRIVATE_TOKEN = 'Qcdm7GJGDdfW95VrpzPc';
const socket = io('http://localhost:3000');

  function handleFormSubmit(event) {⋯

  function getIssues(repositoryId, cb) {⋯

  function getRepositories(cb) {⋯

  function sendMessage(event) {
    event.preventDefault();
    const message = document.getElementById('message').value;
    const repository = document.getElementById('repository').value;
    socket.emit('message',message);
  }

  getRepositories((err, repositories) => {⋯
```

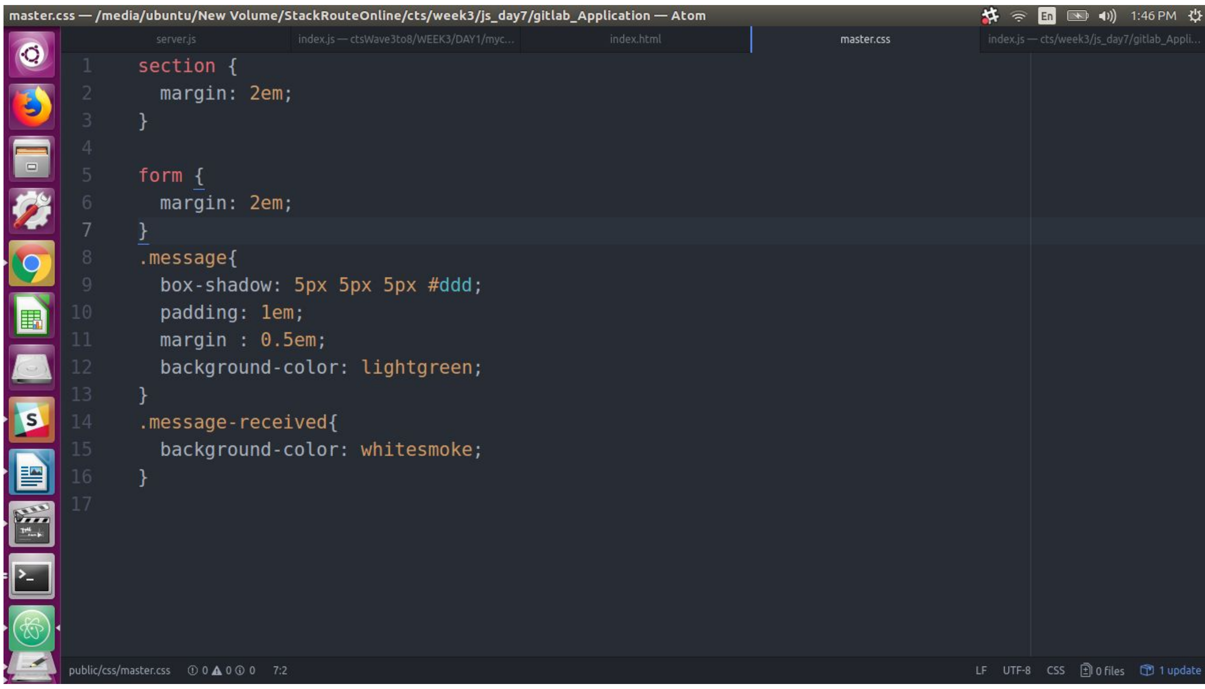
c) See the result.

d) Print the chat on ui


index.js — cts/week3/js_day7/gitlab_Application/public/js — /media/ubuntu/New Volume/StackRouteOnline/cts/week3/js_day7/gitlab_Application —
1:45 PM
server.js
index.js — ctsWave3to8/WEEK3/DAY1/myc...
index.html
master.css
index.js — cts/week3/js_day7/gitlab_Appli...
public/js/index.js
90:23
LF UTF-8 JavaScript
0 files
1 update

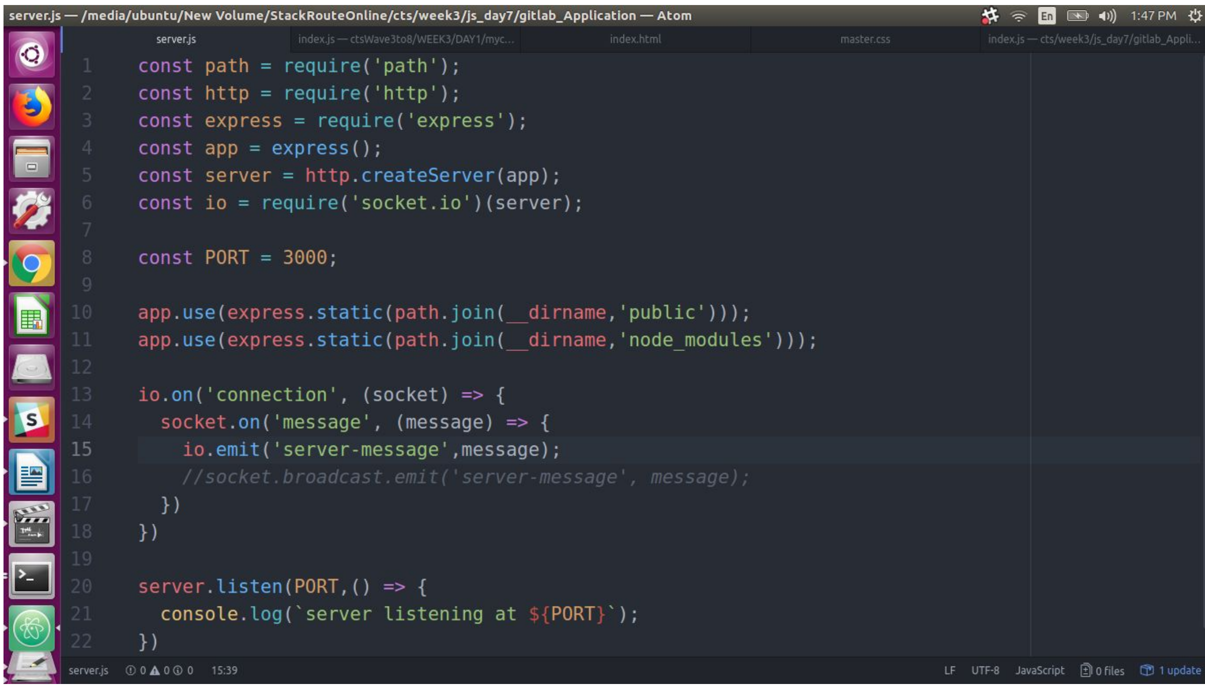
```
function sendMessage(event) {
  event.preventDefault();
  const message = document.getElementById('message').value;
  const repository = document.getElementById('repository').value;

  const messageElement = document.createElement('div');
  messageElement.innerText = message;
  messageElement.className = 'message align-self-start';
  document.getElementById('chatHistory').appendChild(messageElement);
  socket.emit('message',message);
}
socket.on('server-message',(message)=>{
  const messageElement = document.createElement('div');
  messageElement.innerText = message;
  messageElement.className = 'message message-received align-self-start';
  document.getElementById('chatHistory').appendChild(messageElement);
})
getRepositories((err, repositories) => {
  //const repositoryNames = repositories.map(repository => repository.path_with_namespace);
 //socket.emit('message',repositoryNames);
    const tbody = document.getElementsByTagName('tbody')[0];
```

e) Css


master.css — /media/ubuntu/New Volume/StackRouteOnline/cts/week3/js_day7/gitlab_Application — Atom
1:46 PM
server.js
index.js — ctsWave3to8/WEEK3/DAY1/myc...
index.html
master.css
index.js — cts/week3/js_day7/gitlab_Appli...
public/css/master.css
7:2
LF UTF-8 CSS
0 files
1 update

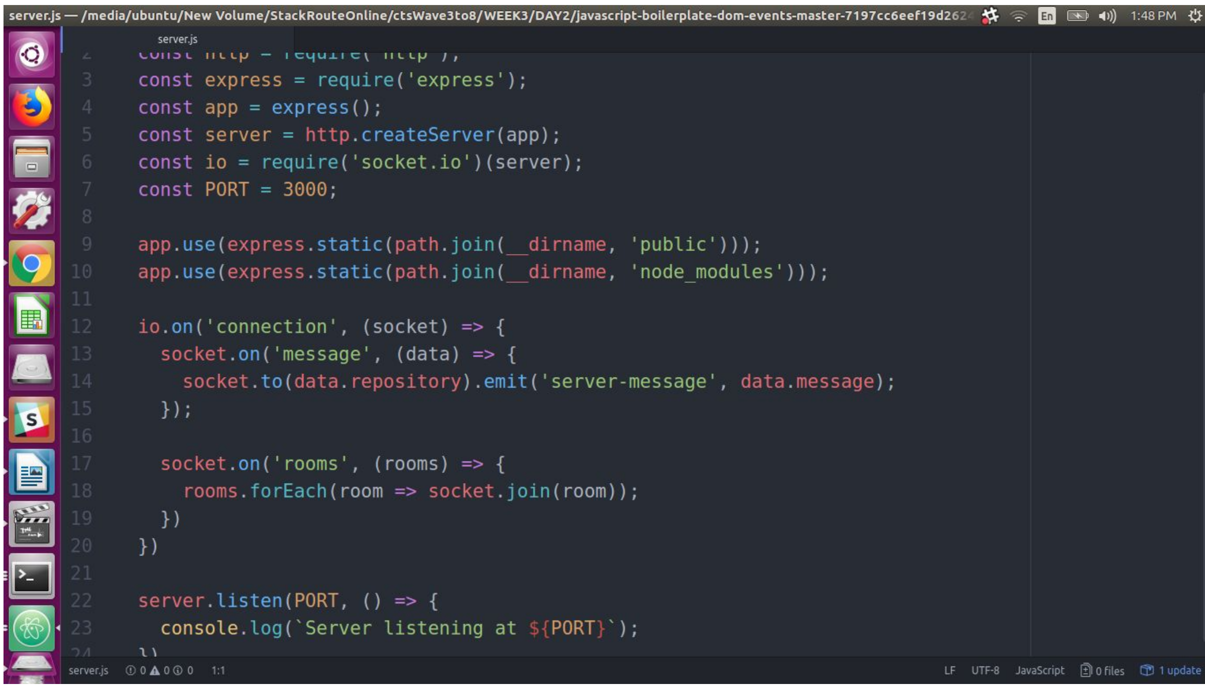
```
section {
  margin: 2em;
}

form {
  margin: 2em;
}
.message{
  box-shadow: 5px 5px 5px #ddd;
  padding: 1em;
  margin : 0.5em;
  background-color: lightgreen;
}
.message-received{
  background-color: whitesmoke;
}
```

f) server.js


server.js — /media/ubuntu/New Volume/StackRouteOnline/cts/week3/js_day7/gitlab_Application — Atom

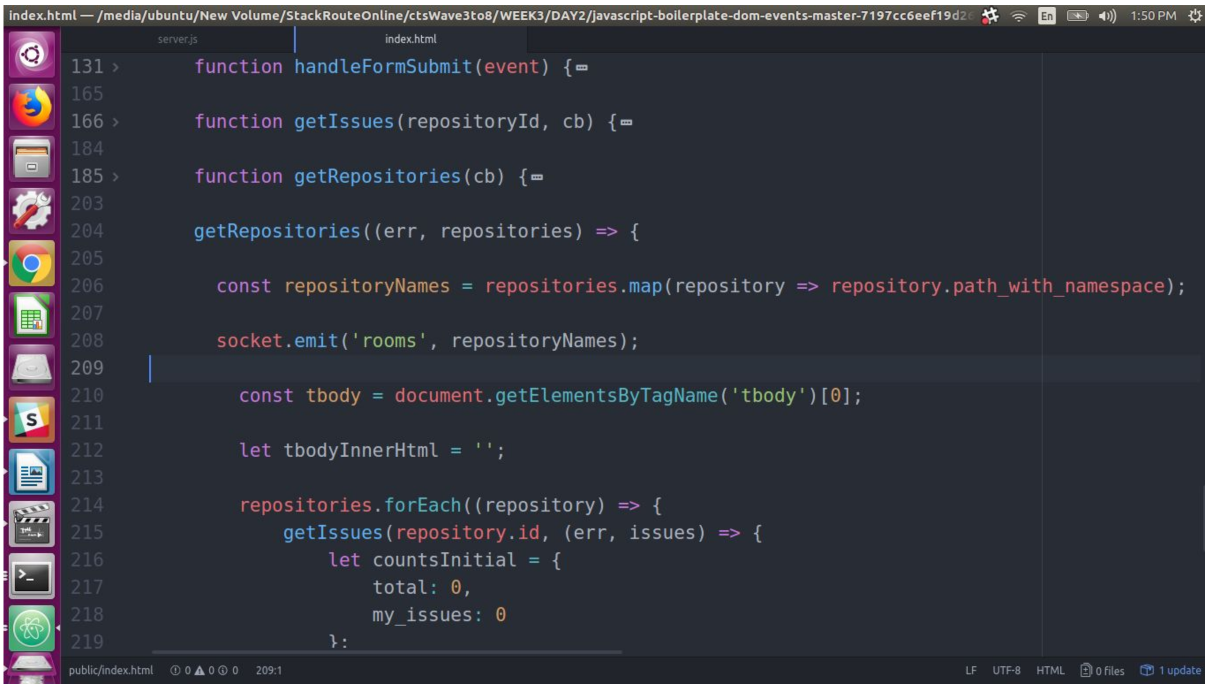
```
const path = require('path');
const http = require('http');
const express = require('express');
const app = express();
const server = http.createServer(app);
const io = require('socket.io')(server);

const PORT = 3000;

app.use(express.static(path.join(__dirname,'public')));
app.use(express.static(path.join(__dirname,'node_modules')));

io.on('connection', (socket) => {
  socket.on('message', (message) => {
    io.emit('server-message',message);
    //socket.broadcast.emit('server-message', message);
  })
})

server.listen(PORT,() => {
  console.log(`server listening at ${PORT}`);
})
```

g) To send message to a room(server.js):-


server.js — /media/ubuntu/New Volume/StackRouteOnline/ctsWave3to8/WEEK3/DAY2/javascript-boilerplate-dom-events-master-7197cc6eef19d262

```
const http = require('http');
const express = require('express');
const app = express();
const server = http.createServer(app);
const io = require('socket.io')(server);
const PORT = 3000;

app.use(express.static(path.join(__dirname, 'public')));
app.use(express.static(path.join(__dirname, 'node_modules')));

io.on('connection', (socket) => {
  socket.on('message', (data) => {
    socket.to(data.repository).emit('server-message', data.message);
  });

  socket.on('rooms', (rooms) => {
    rooms.forEach(room => socket.join(room));
  })
})

server.listen(PORT, () => {
  console.log(`Server listening at ${PORT}`);
})
```

h)



index.html — /media/ubuntu/New Volume/StackRouteOnline/ctsWave3to8/WEEK3/DAY2/javascript-boilerplate-dom-events-master-7197cc6eef19d2

server.js   index.html

```
    function handleFormSubmit(event) {⋯

    function getIssues(repositoryId, cb) {⋯

    function getRepositories(cb) {⋯

    getRepositories((err, repositories) => {

      const repositoryNames = repositories.map(repository => repository.path_with_namespace);

      socket.emit('rooms', repositoryNames);

        const tbody = document.getElementsByTagName('tbody')[0];

        let tbodyInnerHtml = '';

        repositories.forEach((repository) => {
            getIssues(repository.id, (err, issues) => {
                let countsInitial = {
                    total: 0,
                    my_issues: 0
                };
```

public/index.html   0 0 0   209:1   LF   UTF-8   HTML   0 files   1 update