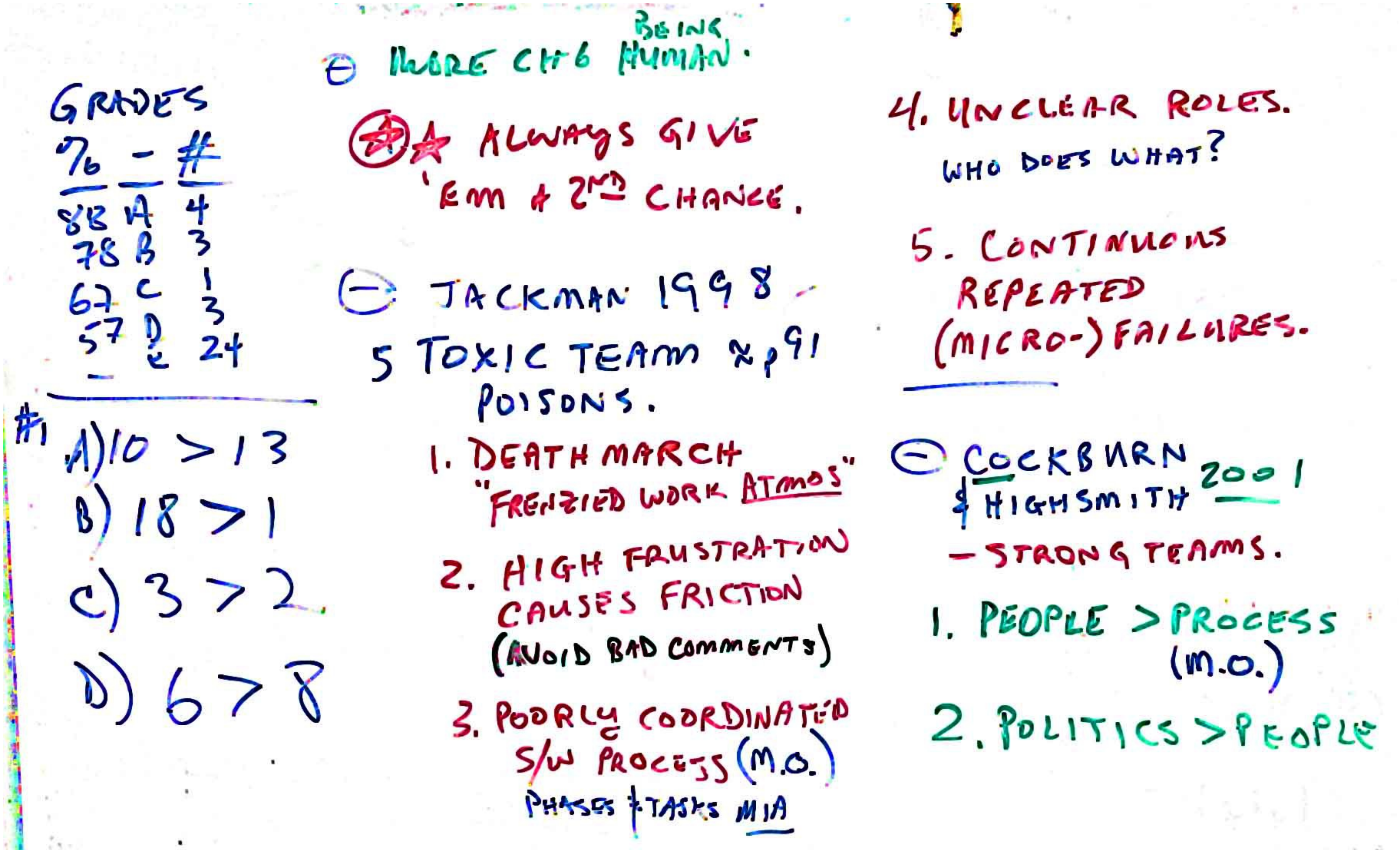
# CECS 343-03/4 — S/W Engr'g — Lecture 2/17/20 Pictures



GRADES

| % | – | # |
|---|---|---|
| 88 | A | 4 |
| 78 | B | 3 |
| 67 | C | 1 |
| 57 | D | 3 |
| – | E | 24 |

#1 A) 10 > 13

B) 18 > 1

C) 3 > 2

D) 6 > 8

⊖ MORE CH 6 BEING HUMAN.

✪ ALWAYS GIVE 'EM A 2ND CHANCE.

⊖ JACKMAN 1998 –

5 TOXIC TEAM & p91 POISONS.

1. DEATH MARCH "FRENZIED WORK ATMOS"
2. HIGH FRUSTRATION CAUSES FRICTION (AVOID BAD COMMENTS)
3. POORLY COORDINATED S/W PROCESS (M.O.) PHASES + TASKS MIA
4. UNCLEAR ROLES. WHO DOES WHAT?
5. CONTINUOUS REPEATED (MICRO-) FAILURES.

⊖ COCKBURN & HIGHSMITH 2001

– STRONG TEAMS.

1. PEOPLE > PROCESS (M.O.)
2. POLITICS > PEOPLE

CH 7 PRINCIPLES
"TO USE".

(GIVE "ACTIONABLE"
ADVICE)

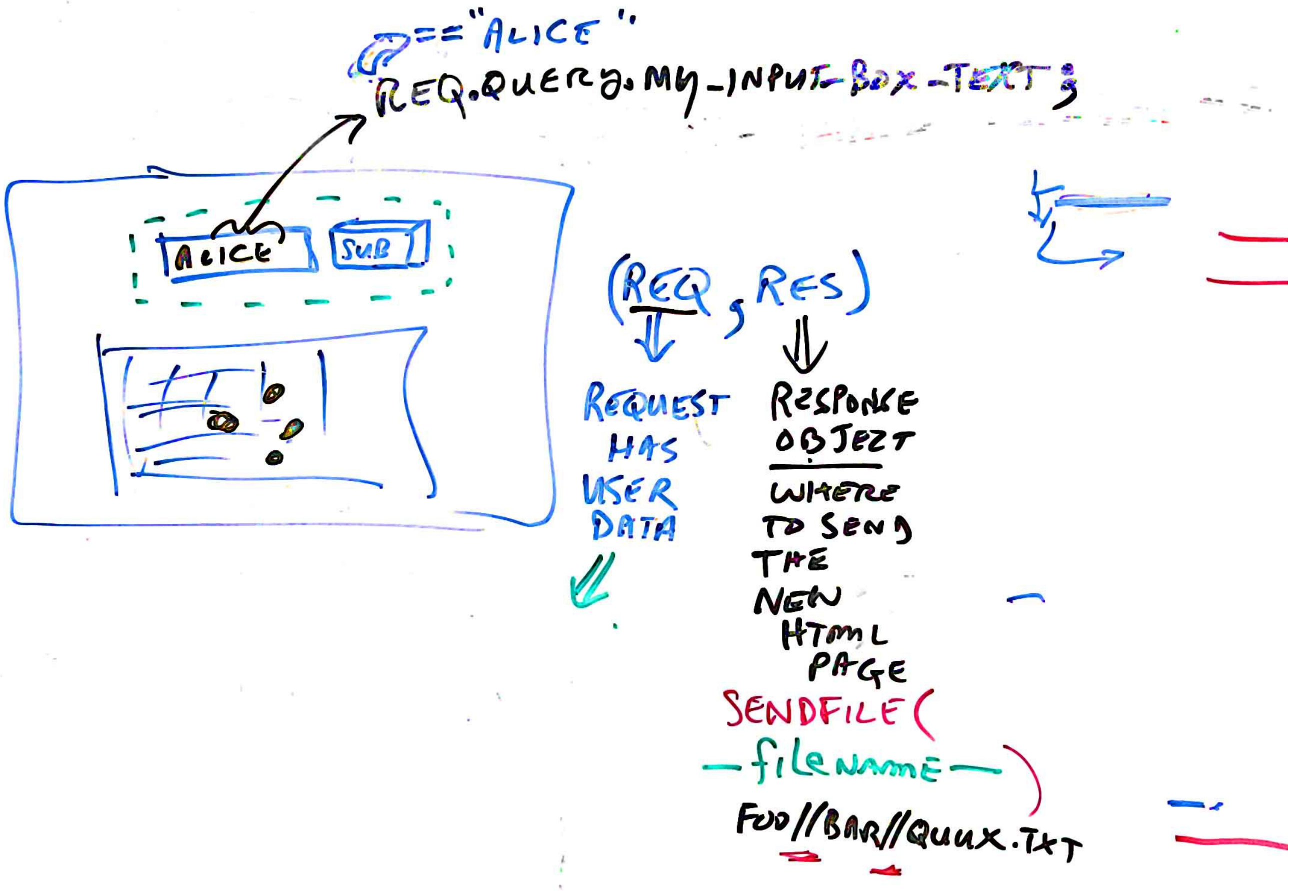
MEAN STACK

M - MYSQL RDB.
E - EXPRESS
A - ANGULAR
N - NODE

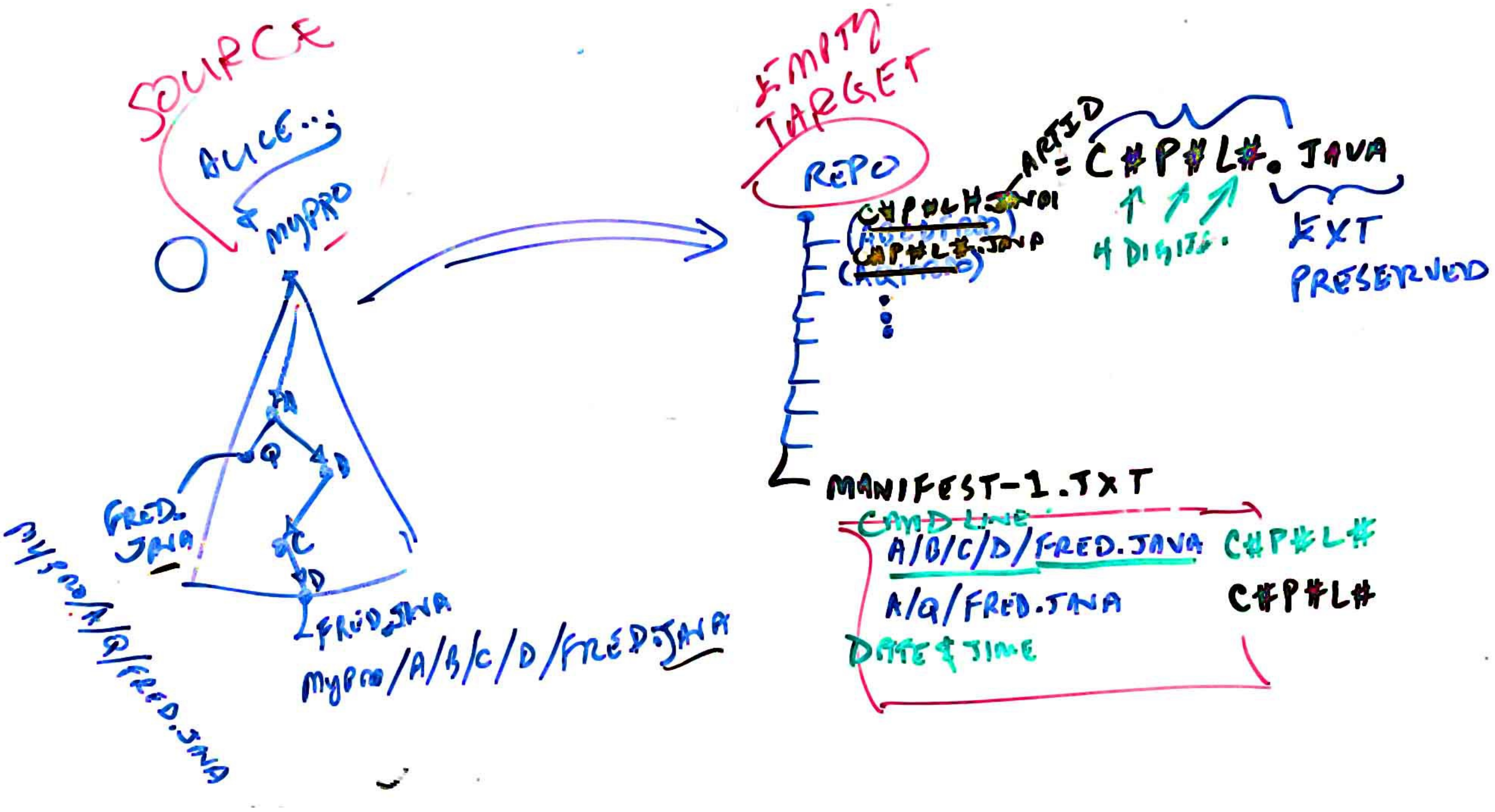
MERN STACK

M -
E -
R - REACT
N -

# CECS 343-03/4 — S/W Engr'g — Lecture 2/17/20 Pictures

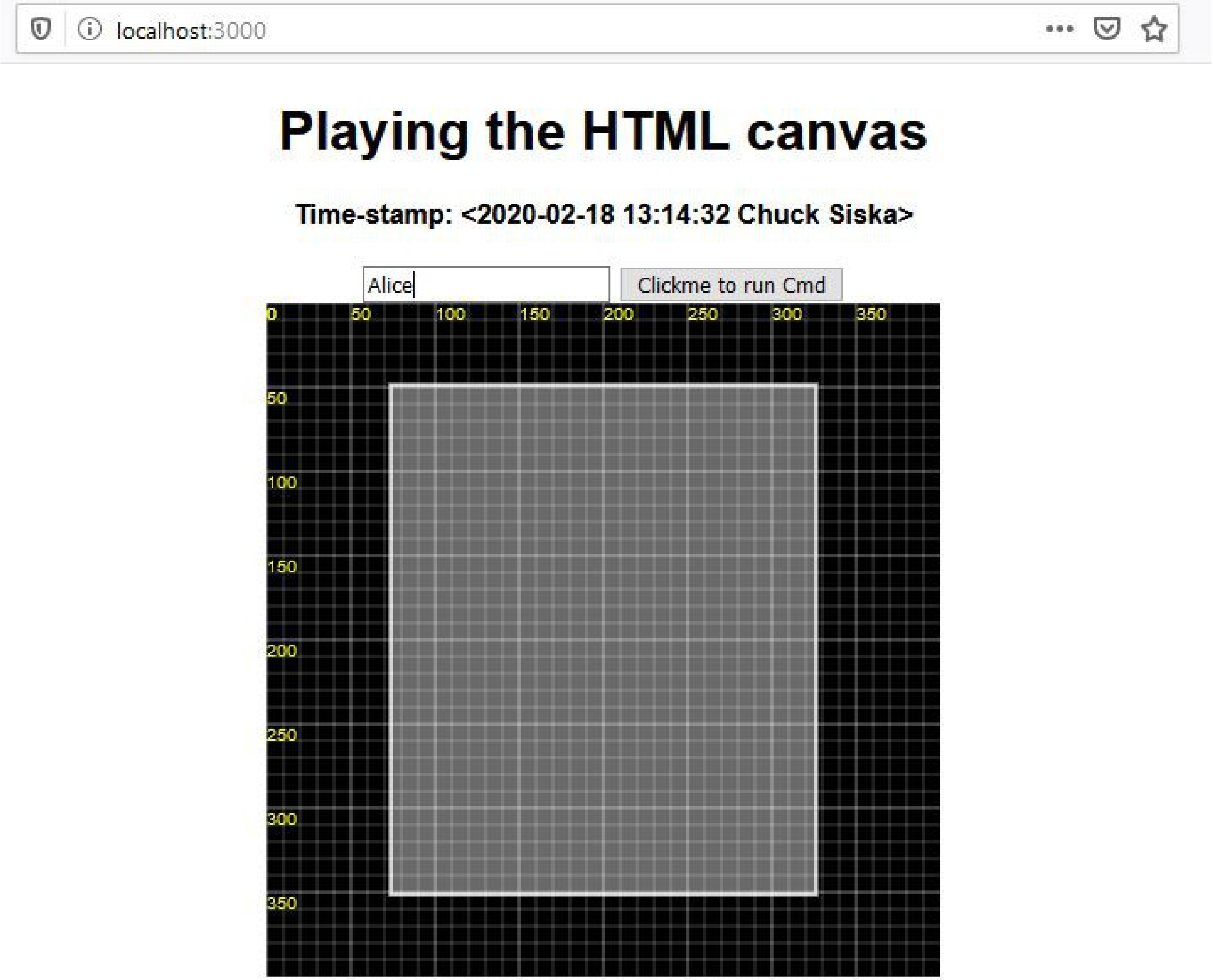


# CECS 343-03/4 — S/W Engr'g — Lecture 2/17/20 Pictures



## CECS 343-03/4 — S/W Engr'g — Lecture 2/17/20 Pictures

Client-Server screenshot, BEFORE clicking the button to submit the input textbox contents



localhost:3000

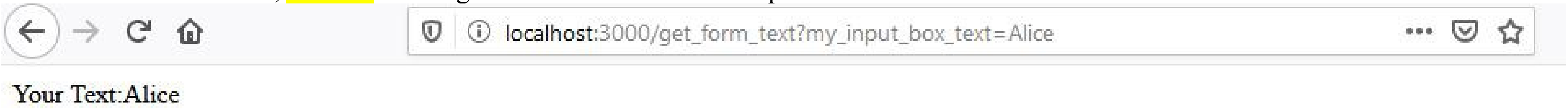
# Playing the HTML canvas

**Time-stamp: <2020-02-18 13:14:32 Chuck Siska>**

Alice

## CECS 343-03/4 — S/W Engr'g — Lecture 2/17/20 Pictures

Client-Server screenshot, AFTER clicking the button to submit the input textbox contents



localhost:3000/get_form_text?my_input_box_text=Alice

Your Text:Alice

Here's the HTML file

```
<!doctype html><html><head>
<title>This is an HTML canvas</title>
<link rel="stylesheet"
href="assets/styles.css">
</head>
<body><h1>Playing the HTML canvas</h1>
<h4>Time-stamp: <2020-02-18 13:14:32 Chuck
Siska></h4>

<form action="/get_form_text" method="GET">
    <input type="text" id="box_1" cols="55"
       name="my_input_box_text" required />
    <input type="submit"
          value="Clickme to run Cmd" />
</form>

<canvas id="grid" width="400"
        height="400"></canvas>

</body>
<script type="text/javascript"
        src="assets/draw-stuff.js"></script>
<script>

// after "Introducing JavaScript Game Development" by
Stuart.

// ================================  draw_title ====
// Draw filled rect.
function draw_title( context )
{
    context.save( );
    context.fillStyle = 'lightgrey';
    context.font = "30px Arial";
    context.fillText( "Grid", 150, 140 );
    context.restore( );
}
// ======================================  Main ====
  var canvas = document.getElementById( "grid" );
  var context = canvas.getContext( "2d" );
  draw_rect( context );
  // Draw a grid
  draw_grid( context, 10, 50, 'white', 'yellow' );
  //draw_title( context );

</script>
</html>
```

Here's the server-side Node-Express file.

```
// File: app.js
// Time-stamp: <2020-02-18 13:26:13 Chuck Siska>

var express = require('express');
var app = express();  // Init an Express object.
    /// This is used to avoid problems 'loading' JS and CSS files, etc.
    app.use( express.static( './' )); //Avoid 'MIME type ("text/html") mismatch'

    /// This is run when you click on the button in the form with 'action="/get_form_text"'.
    app.get( // Handle a client-side action request.
        '/get_form_text', // For this URL sub-tag action:
        function(req, res){  // Run this fcn.  /// This function builds the response 'webpage'.
        // Get the text from the URL request packet.
        var myText = req.query.my_input_box_text; /// This is where you get the user's input text.
        console.log('App.js rcvd = ' + myText + '.');
        // Reply to the client GUI using that text.
        res.send('Your Text:' +myText);  /// This 'builds' and sends the response 'webpage'.
    });

/// This is run when you got to the http://localhost:3000/ webpage.
app.get('/', function (req, res) { // Set page-gen fcn for URL root request.
    //Old: res.send('Hello World!'); // Send webpage containing "Hello World!".
    // NB, need absolute path here (including correct drive location).
    res.sendFile( 'G://csu//assets//js-node//myapp//js-1-edt.html' ); /// The initial webpage.
});

app.listen(3000, function () { // Set callback action fcn on network port.
    console.log('App.js listening on port 3000!'); /// See this after you do >node app.js

});
```