[**Web Application Development Needs**](#_332hda9hlbw) **1**

[**.npmrc**](#_2wde242o19vu) **2**

[**.browserslistrc**](#_qyk1ip5y83qu) **3**

[**Bootstrap**](#_6yz6xryod7nd) **3**

[**Cloud MongoDB Details**](#_vw2m4xya5vv5) **3**

[**REST API Security Essentials**](#_nroqitwn7e84) **3**

[**Secure a REST API using JWT**](#_6yk75biwhu6u) **4**

[**Linux Commands**](#_aic0g1ksferj) **5**

[**Delete the local branch**](#_g9cerbkc1bd2) **5**

[**Heroku Account**](#_8xji11gaohbr) **6**

[**Description**](#_eeo4d67z17op) **6**

[**Functional requirements**](#_1sx6jnm4roet) **6**

[**Style requirements**](#_etbnhoj6aj0n) **7**

[**Examples**](#_ncbilgc8l31h) **7**

[**Libraries**](#_ktzcsvc8mgq8) **8**

## Web Application Development Needs

| Technology | Example | Npms |
| --- | --- | --- |
| HTML5, CSS3, ES6 | Will works with latest supported browsers |  |
| -UI Frameworks | 1. Angular Material 2. Bootstrap 5 3. Primeng 4. [Ionic](https://ionicframework.com/docs/components) 5. [cordova](https://cordova.apache.org/plugins/) |  |
| Micro Frontend Applications | Single Spa Applications | 1. single-spa 2. systemjs |
| Modern JavaScript Frameworks | These modern javascript frameworks developed in JavaScript and typescript. These are allow us to provide the util methods take care of browser agnostic and performance | 1. loadjs 2. babel 3. rxjs 4. ngx 5. ngrx 6. zone.js |
| Server Side JavaScript Frameworks | This works like Http Web Server. | 1. [nodejs](https://nodejs.org/en/) 2. nextjs 3. [koa](https://www.npmjs.com/package/koa) |
| Security for REST APIs |  | 1. cookie-parser 2. body-parser 3. csurf 4. jwt-simple 5. helmet 6. passport 7. ssl-root-cas |
| Provide SSL |  | 1. ssl-root-cas |
| UI Grids / Tables |  | 1. datatables 2. ag-grid |
| UI Build Tools |  | 1. gulp 2. webpack |
| Logging |  | 1. bunyan |
| REST APIs |  | 1. request 2. http 3. jio |
| Code Editor |  | 1. ace-build |
| Charts |  | 1. highchart 2. keylines(paid) |

## .npmrc

This is used for provide the additional lookup for npm libraries for downloading and install

## .browserslistrc

This is used for provide the list of supported browsers

## Bootstrap

1. Grid System
2. Responsive UI components
3. Highly Flexible
4. Ease Of Use
5. Comprehensive List of Components
6. Leveraging JS Libraries
   1. Tabs, ScrollSpy, Tooltips, Pop Overs, Alerts

## Cloud MongoDB Details

Login: [https://cloud.mongodb.com/](https://cloud.mongodb.com/v2/621193978ad1847d9c144790#clusters)

Login with Google Account with username and password

DB Username: sateesh

Password: KYZmMF9EEhurKrz8

Database name: sateesh-app

Connect URL: mongodb+srv://sateesh:KYZmMF9EEhurKrz8@sateesh-app.as2kh.mongodb.net/myFirstDatabase?retryWrites=true&w=majority

mongodb+srv://sateesh:KYZmMF9EEhurKrz8@sateesh-app.as2kh.mongodb.net/stock-db?retryWrites=true&w=majority

## REST API Security Essentials

There are multiple ways to secure a RESTful API e.g. [basic auth](http://howtodoinjava.com/resteasy/jax-rs-resteasy-basic-authentication-and-authorization-tutorial/), [OAuth](https://oauth.net/) etc. but one thing is sure that RESTful APIs should be stateless – so request authentication/authorization should not depend on cookies or sessions. Instead, each API request should come with some sort authentication credentials which must be validated on the server for each and every request.

REST Security Design Principles

1. Least Privilege
2. Fail-Safe Defaults
3. Economy of Mechanism
4. Complete Mediation
5. Open Design
6. Separation of Privilege:
7. Least Common Mechanism
8. Psychological Acceptability

Best Practices to Secure REST APIs

1. Keep it Simple
2. Always Use HTTPS
3. Use Password Hash
4. Never expose information on URLs
5. Consider Oauth
6. Consider Adding Timestamp in Request
7. Input Parameter Validation

## Secure a REST API using JWT

[JWT](https://blog.logrocket.com/how-to-secure-a-rest-api-using-jwt-7efd83e71432/) Stands for JSON Web Tokens

What is a JWT

JSON Web Tokens are an open and standard ([RFC 7519](https://tools.ietf.org/html/rfc7519)) way for you to represent your user’s identity securely during a two-party interaction. That is to say, when two systems exchange data you can use a JSON Web Token to identify your user without having to send private credentials on every request.

Client app 
I. Son 
Returns 

1. The user/client app sends a sign in request. In other words, here is where your username/password (or any other type of sign in credentials you need to provide) will travel
2. Once verified, the API will create a JSON Web Token (more on this in a bit) and sign it using a secret key
3. Then the API will return that token back to the client application
4. Finally, the client app will receive the token, verify it on its own side to make sure it’s authentic and then proceed to use it on every subsequent request to authenticate the user without having to send their credentials anymore

1. The structure of the token : header.payload.signature
2. Using the token

## Linux Commands

**#####**

**# Create Log dir inside micro services**

**find . -maxdepth 1 -type d -exec mkdir {}/log \;**

**####**

**# Delete Node modules inside micro services**

**find . -name \"node\_modules\" -maxdepth 2 -type d | xargs rm -rf**

**######**

**# Delete only cloudnit modules inside node\_modules**

**find . -name "cloudknit\*" -maxdepth 3 -type d | xargs rm -rf**

## Delete the local branch

[**https://blog.mdavid626.com/2019/11/17/How-to-delete-all-local-branches-except-master-in-git/**](https://blog.mdavid626.com/2019/11/17/How-to-delete-all-local-branches-except-master-in-git/)

**git branch --format '%(refname:lstrip=2)' | Where-Object { $\_ -ne 'master' } | ForEach-Object { git branch -d $\_ }**

## .bashrc

alias **ll**='ls -lrt'

#export DB\_HOST=10.193.21.21

#export DB\_HOST=10.193.19.110

#export DB\_HOST=10.193.17.99 # Vamsi

#export DB\_HOST=10.193.19.81 # Sai

#export DB\_HOST=10.193.16.244 # Sateesh OpenLab

#export DB\_HOST=10.193.16.233 # Atish DB - salesforcetestapril18@gmail.com - salesForce123

#export DB\_HOST=10.193.68.170 # Andy DB for 10.193.6.127:

#export DB\_HOST=10.193.68.172 # Andy DB for 10.193.6.127:

#export DB\_HOST=10.193.68.169 # Andy DB for 10.193.6.127:

#export DB\_HOST=10.193.16.168

#export DB\_HOST=10.193.16.184 # Bansi SFDC Stack

#export DB\_HOST=10.193.18.85 # Scale setup

#export DB\_HOST=10.193.18.184 # Janet

#export DB\_HOST=10.193.16.78

#export DB\_HOST=10.193.16.71

#export DB\_HOST=10.193.16.186

#export DB\_HOST=10.193.161.242

#export DB\_HOST=10.193.6.38

#export DB\_HOST=10.193.17.202

#export DB\_HOST=10.193.17.225 # Sai QA Server

#export DB\_HOST=10.193.18.146

export **PROTOCOL**=https

export **CALLBACK\_HOST**=localhost:8000

export **cloudcontrolapigateway**=https://localhost:3000

export **PATH**="/usr/local/opt/mongodb-community@4.0/bin:$PATH"

# Setting PATH for Python 2.7

# The original version is saved in .bash\_profile.pysave

**PATH**="/Library/Frameworks/Python.framework/Versions/2.7/bin:${PATH}:/Users/sateesh/Documents/RBTools"

export PATH

**P4CONFIG**="/Users/sateesh/.p4config"

export **STACK\_REGION**="Germany"

#export SALESFORCE\_CLIENT\_ID="3MVG9d8..z.hDcPJWD.JaNZeMPLn4z8dEW6WmdMp25Qu2NHoIQN4ZIkT9EzIM\_hB11sSvH\_MI\_NavOBEN5EmC"

alias **gitls**='git branch -r | xargs -L1 git --no-pager show -s --oneline --author="$(git config user.name)"'

alias **git\_delete\_branch**='git branch | grep -v "master" | xargs git branch -D'

alias **chrome**='/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --ignore-certificate-errors &> /dev/null &'

alias **delete\_node\_modules**='find . -name "node\_modules" -type d -prune -exec rm -rf "{}" +'

#export AZURE\_CLIENT\_ID=f6c652b8-2f2d-4f27-913b-83c837705edb;

#export AZURE\_CLIENT\_SECRET=-Xvh6nwM\_669.344\_Cv8~.UKeqXOH3TFX3

#export SECONDARY1\_DB\_HOST=10.193.68.169

#export SECONDARY2\_DB\_HOST=10.193.68.172

#export DB\_PORT=27017

#export DB\_WRITE\_CONCERN=3

alias **ssv**=/usr/local/bin/ssv.sh

#alias start-mongodb='/usr/local/opt/mongodb-community@4.0/bin/mongod --sslMode requireSSL --sslPEMKeyFile /Users/sateesh/mongo-certs/mongodb.pem --sslCAFile /Users/sateesh/mongo-certs/ca-fullchain.pem --dbpath /Users/sateesh/mongoDB\_Data-4.0 --bind\_ip 0.0.0.0'

alias **start-mongodb**='/usr/local/opt/mongodb-community@4.0/bin/mongod --dbpath /Users/sateesh/mongoDB\_Data-4.0-without-SSL --bind\_ip 0.0.0.0 -u admin -p 123456'

## Heroku Account

### **Description**

**Implement a progress bar using Angular 6. You should only modify the TS tab, and not the HTML or CSS tab. The task also contains several read-only tests, so consider this a TDD task where you need to implement the solution to pass the prewritten tests.**

### **Functional requirements**

**Create a form with four fields related to a document that is being submitted. There should be a progress indicator at the top that moves forward as you fill each field. The indicator should automatically set the state of the progress bar based on how many of the input fields contain a valid value.**

**The four fields in the document should be the following:**

1. **An input field for the document name with id = "documentName" (this id is used in the tests, so it's important to use this exact identifier);**
2. **A dropdown for the document type with id = "documentType".  
   The dropdown should have three options: empty value (initial state), 'Plain', and 'PDF'.  
   Usage example: <option value="Plain">Plain</option> for Plain, <option value=""></option> for empty value.**
3. **A dropdown for the document category with id = "category".  
   The dropdown should have four options with the following values: empty value (initial state), 'Audit', 'Application', and 'Other' respectively.  
   Usage example: <option value="Audit">Audit</option>.**
4. **An input field for the sender email with id = "email".**

**Here are the functional requirements for the fields validation:**

* **The value of any dropdown field is considered valid if it is not empty;**
* **The document name is considered valid if its length is between 2 and 32 characters, inclusive;**
* **The sender email is considered valid if all the requirements below are satisfied:**
  + **It contains a string that is at most 128 characters long;**
  + **The string matches the format <username>@<domain>, where both username and domain can contain only lowercase English letters, digits, and periods (.);**
  + **Additionally, domain should contain at least one period character (.);**
  + **Both username and domain must not contain any consecutive periods (..).**

**The state of the progress bar indicator should be the following:**

* **0% if all fields values are empty, as they are in the initial state;**
* **25% if only one field has a valid value;**
* **50% for two filled fields;**
* **75% for three valid fields**
* **100% if all of the fields have values selected and all the values are valid.**

### **Style requirements**

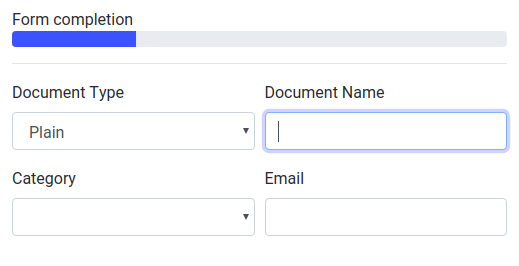
**The style requirements for progress bar are the following:**

* **It should be implemented as two nested divs:**
  + **A div with className = "form-progress-bar-wrapper" of grey color ( #e8ebee) with full screen width;**
  + **A div with className = "form-progress-bar" of blue color ( #007bff) with width reflecting the number of valid values.**
* **The width of the progress bar should be indicating the current filling values: 0%, 25%,50%, 75%, and 100% of the wrapper's width respectively.**

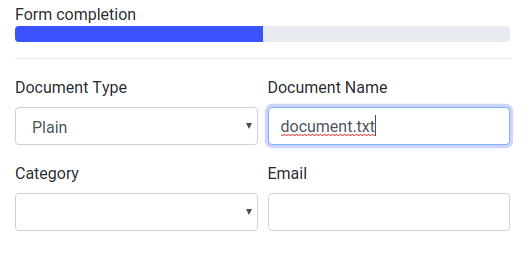
**Feel free to design the progress bar page as you wish, there are no additional requirements for the styling except the ones above.**

### **Examples**

**After selecting the document type the form may look like this:**

****

**And after filling the document name, the form may look like this:**

****

### **Libraries**

**To enable Angular 6.0.5 the following libraries should be included in "Manage Libraries" section (the order matters!):**

* [**https://cdnjs.cloudflare.com/ajax/libs/rxjs/6.2.1/rxjs.umd.min.js**](https://cdnjs.cloudflare.com/ajax/libs/rxjs/6.2.1/rxjs.umd.min.js)
* [**https://cdnjs.cloudflare.com/ajax/libs/core-js/2.5.7/core.js**](https://cdnjs.cloudflare.com/ajax/libs/core-js/2.5.7/core.js)
* [**https://unpkg.com/@angular/core@6.0.5/bundles/core.umd.js**](https://unpkg.com/@angular/core@6.0.5/bundles/core.umd.js)
* [**https://cdnjs.cloudflare.com/ajax/libs/zone.js/0.8.26/zone.min.js**](https://cdnjs.cloudflare.com/ajax/libs/zone.js/0.8.26/zone.min.js)
* [**https://cdnjs.cloudflare.com/ajax/libs/rxjs/6.2.1/rxjs.umd.min.js**](https://cdnjs.cloudflare.com/ajax/libs/rxjs/6.2.1/rxjs.umd.min.js)
* [**https://unpkg.com/@angular/compiler@6.0.5/bundles/compiler.umd.js**](https://unpkg.com/@angular/compiler@6.0.5/bundles/compiler.umd.js)
* [**https://unpkg.com/@angular/platform-browser@6.0.5/bundles/platform-browser.umd.js**](https://unpkg.com/@angular/platform-browser@6.0.5/bundles/platform-browser.umd.js)
* [**https://unpkg.com/@angular/platform-browser-dynamic@6.0.5/bundles/platform-browser-dynamic.umd.js**](https://unpkg.com/@angular/platform-browser-dynamic@6.0.5/bundles/platform-browser-dynamic.umd.js)
* [**https://unpkg.com/@angular/forms@6.0.5/bundles/forms.umd.js**](https://unpkg.com/@angular/forms@6.0.5/bundles/forms.umd.js)

**Feel free to add the above libraries manually if they are not there or add more libraries if you need them.**

***Note: feel free to disable error highlighting within our IDE settings to avoid highlighting issues.***

* **[execution time limit] 10 seconds**

**App**

**1) should load the page**

**2) should display empty form fields**

**3) should fill progress bar by 25% when only documentName input is filled**

**4) should fill progress bar by 50% when documentName and email inputs are filled**

**5) should fill progress bar by 75% when documentName and email inputs are filled and documentType is selected**

**6) should fill progress bar by 100% when all form fields are filled and valid**

**7) should fill progress bar by 25% when documentType select is empty, category is selected, documentName and email inputs are invalid**

**8) should fill progress bar by 75% when both documentType and category are selected, documentName is filled and email input is invalid**

**0 passing (5s)**

**8 failing**

**1) App**

**should load the page:**

**Error: Timeout of 5000ms exceeded. For async tests and hooks, ensure "done()" is called; if returning a Promise, ensure it resolves. (/usercode/FILESYSTEM/specs/simple.test.js)**

**at listOnTimeout (node:internal/timers:557:17)**

**at processTimers (node:internal/timers:500:7)**

**2) App**

**should display empty form fields:**

**Error: Error: failed to find element matching selector "select[id="category"]"**

**at ElementHandle.$eval (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/JSHandle.js:805:19)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:20:12)**

**3) App**

**should fill progress bar by 25% when only documentName input is filled:**

**TypeError: Cannot read properties of null (reading 'click')**

**at setDocumentNameInputValue (specs/simple.test.js:111:17)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:27:5)**

**4) App**

**should fill progress bar by 50% when documentName and email inputs are filled:**

**TypeError: Cannot read properties of null (reading 'click')**

**at setEmailInputValue (specs/simple.test.js:117:17)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:38:5)**

**5) App**

**should fill progress bar by 75% when documentName and email inputs are filled and documentType is selected:**

**Error: No node found for selector: select[id="documentType"]**

**at assert (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/assert.js:26:15)**

**at DOMWorld.select (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/DOMWorld.js:303:32)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:49:5)**

**6) App**

**should fill progress bar by 100% when all form fields are filled and valid:**

**Error: No node found for selector: select[id="category"]**

**at assert (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/assert.js:26:15)**

**at DOMWorld.select (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/DOMWorld.js:303:32)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:60:5)**

**7) App**

**should fill progress bar by 25% when documentType select is empty, category is selected, documentName and email inputs are invalid:**

**Error: No node found for selector: select[id="category"]**

**at assert (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/assert.js:26:15)**

**at DOMWorld.select (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/DOMWorld.js:303:32)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:73:5)**

**8) App**

**should fill progress bar by 75% when both documentType and category are selected, documentName is filled and email input is invalid:**

**Error: No node found for selector: select[id="category"]**

**at assert (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/assert.js:26:15)**

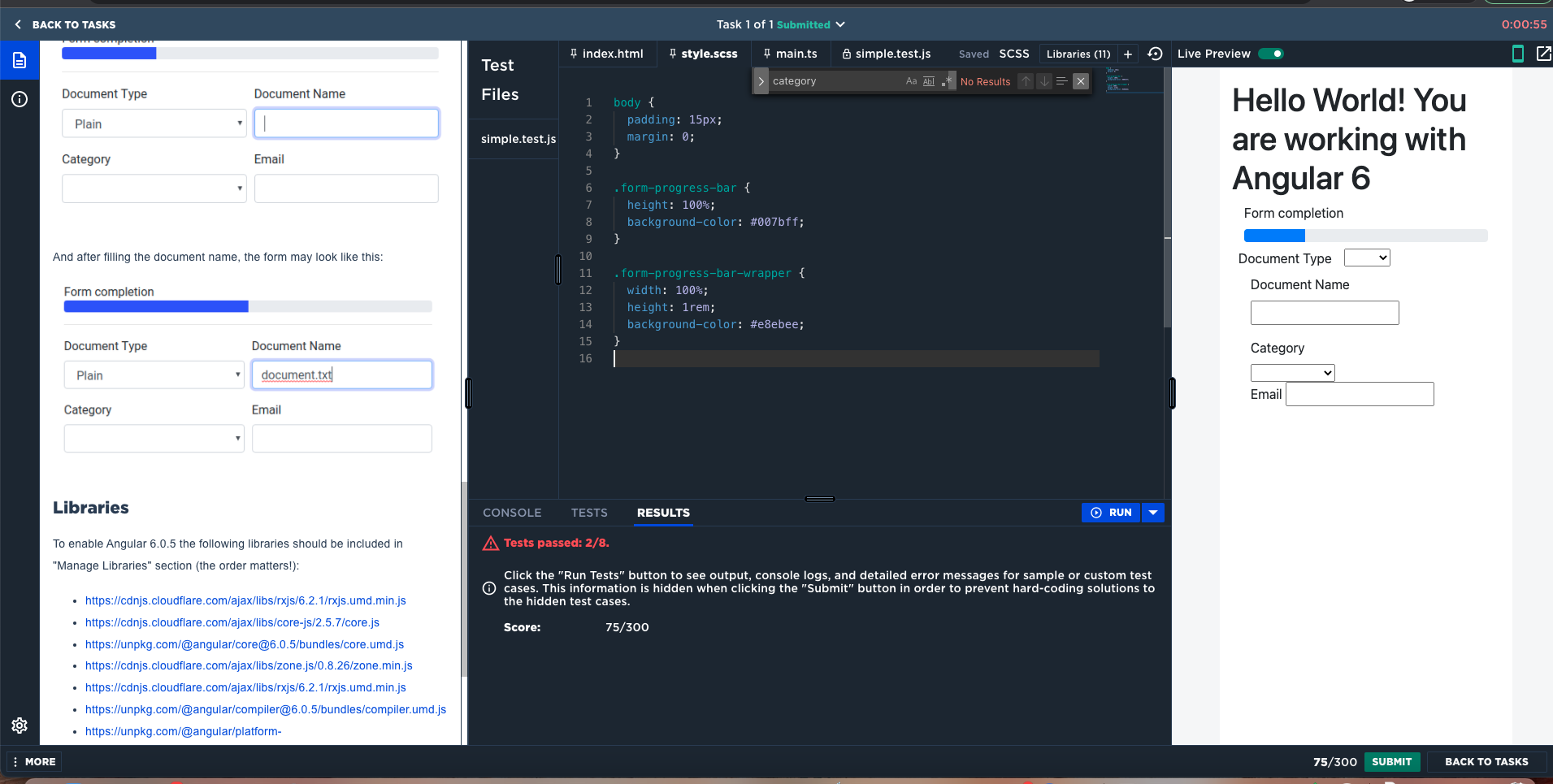
**at DOMWorld.select (/root/.nvm/versions/node/v16.13.0/lib/node\_modules/puppeteer/lib/cjs/puppeteer/common/DOMWorld.js:303:32)**

**at processTicksAndRejections (node:internal/process/task\_queues:96:5)**

**at async Context.<anonymous> (specs/simple.test.js:89:5)**

**Click the "Run Tests" button to see output, console logs, and detailed error messages for sample or custom test cases. This information is hidden when clicking the "Submit" button in order to prevent hard-coding solutions to the hidden test cases.**

**Score:75/300**

****