

# Test Results Viewer

## 1. Steps to run the project:

Prerequisites: nodejs and git applications should be installed on your computer

### **i. Clone the project repository**

Open a directory where you want to clone the repository in the command prompt or MS visual code. Then run following command:

Git clone <https://github.com/bharatgpatil/test-results-viewer.git>

```
G:\work\software engineering\project>Git clone https://github.com/bharatgpatil/test-results-viewer.git
Username for 'https://github.com': bharatgpatil
Password for 'https://bharatgpatil@github.com':
remote: Enumerating objects: 1725, done.
remote: Counting objects: 100% (1725/1725), done.
remote: Compressing objects: 100% (1511/1511), done.
remote: Total 1725 (delta 175), reused 1725 (delta 175), pack-reused 0
Receiving objects: 100% (1725/1725), 19.44 MiB | 2.47 MiB/s, done.
Resolving deltas: 100% (175/175), done.
Checking out files: 100% (1698/1698), done.
```

### **ii. Install Dependencies:**

Got to the project folder in the command prompt or MS visual code as shown below:

Run '[npm install](#)' command to install all project dependencies.

```
G:\work\software engineering\project>cd test-results-viewer

G:\work\software engineering\project\test-results-viewer>npm install
npm WARN deprecated @angular/http@7.2.4: Switch to @angular/common/http - see https://angular.io/guide/http
npm WARN deprecated core-js@2.6.4: core-js@<3.0 is no longer maintained and not recommended for usage due to the
npm WARN deprecated istanbul@0.4.5: This module is no longer maintained, try this instead:
npm WARN deprecated   npm i nyc
```

### **iii. Development server:**

Run '[npm start](#)' for running frontend and backend server concurrently.

```
G:\work\software engineering\project\test-results-viewer>npm start

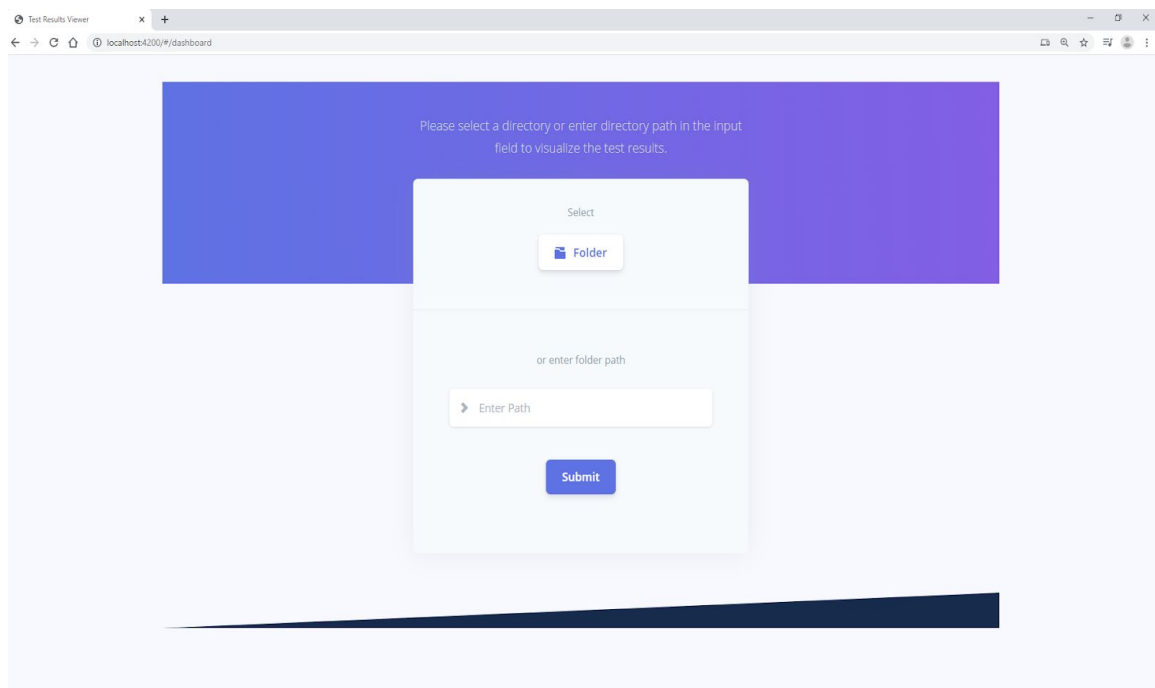
> test-results-viewer@1.0.1 start G:\work\software engineering\project\test-results-viewer
> concurrently "npm run frontend" "npm run server"

[0] > test-results-viewer@1.0.1 frontend G:\work\software engineering\project\test-results-viewer
[0] > ng serve
[0]
[1] > test-results-viewer@1.0.1 server G:\work\software engineering\project\test-results-viewer
[1] > nodemon src/backend/app.js
[1]
[1] [nodemon] 2.0.2
[1] [nodemon] to restart at any time, enter `rs`
[1] [nodemon] watching dir(s): *.*
[1] [nodemon] watching extensions: js,mjs,json
[1] [nodemon] starting `node src/backend/app.js`
[1] Sample API is running on 4400
[0] ** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
[0]
[0] Date: 2020-01-25T17:39:33.418Z
[0] Hash: 82788daa24e6ce59ac12
[0] Time: 15027ms
[0] chunk {layouts-main-layout-main-layout-module} layouts-main-layout-main-layout-module.js, layouts-main-layout-main-
[0] chunk {main} main.js, main.js.map (main) 50.2 kB [initial] [rendered]
[0] chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 237 kB [initial] [rendered]
[0] chunk {runtime} runtime.js, runtime.js.map (runtime) 8.82 kB [entry] [rendered]
[0] chunk {scripts} scripts.js, scripts.js.map (scripts) 166 kB [entry] [rendered]
[0] chunk {styles} styles.js, styles.js.map (styles) 1.42 MB [initial] [rendered]
[0] chunk {vendor} vendor.js, vendor.js.map (vendor) 5.39 MB [initial] [rendered]
[0] i [wdm]: Compiled successfully.
```

frontend dev server runs on the port 4200 and backend dev server uses port 4400. (you can also run them separately by using '*npm run frontend*' and '*npm run server*' commands).

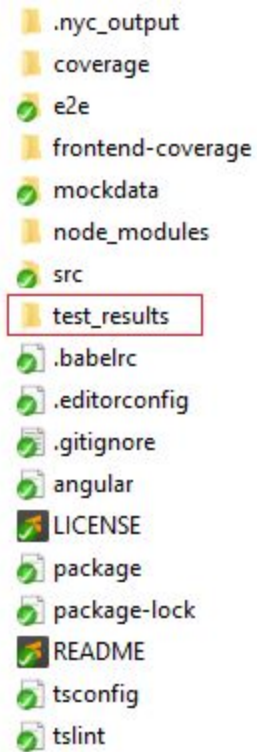
#### iv. Run the project:

Open <http://localhost:4200/> in the browser



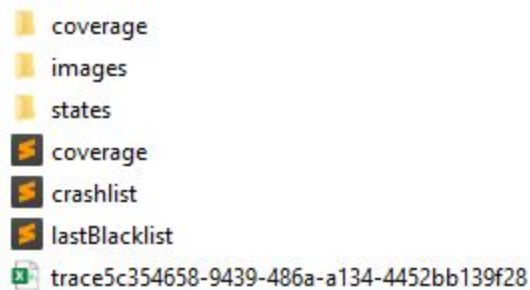
## 2. Deploying test data:

Create a folder called 'test\_results' in the project source code directory as shown below.



Then create 'public' folder inside the test\_results

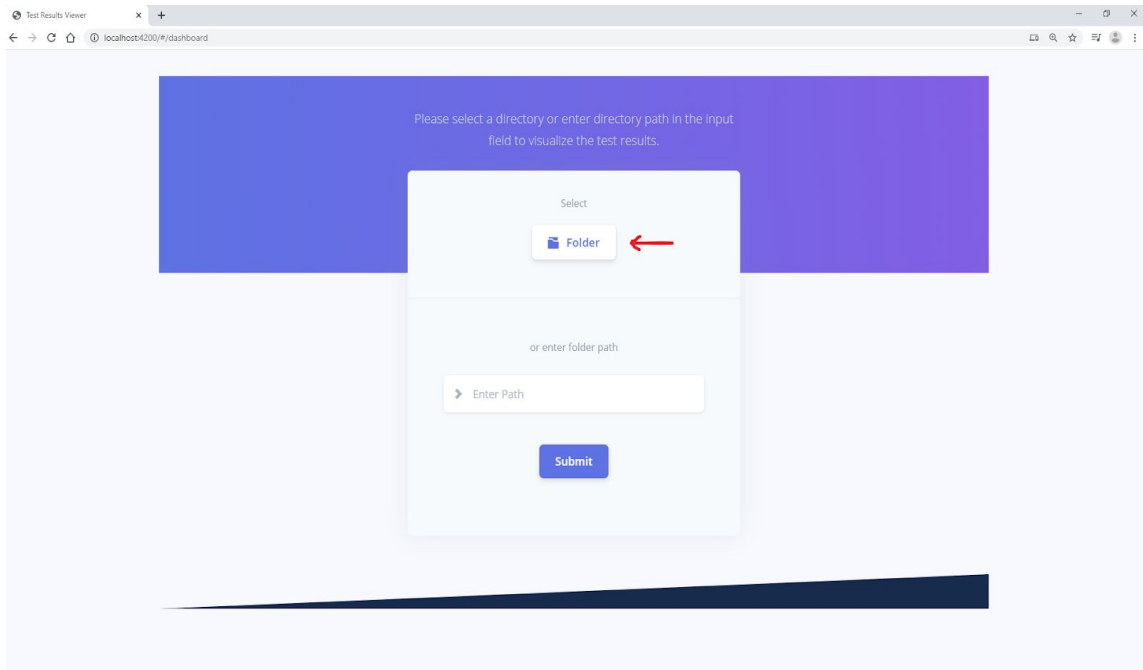
Now create 'data' directory inside 'public' folder and copy the test result data in the following way:



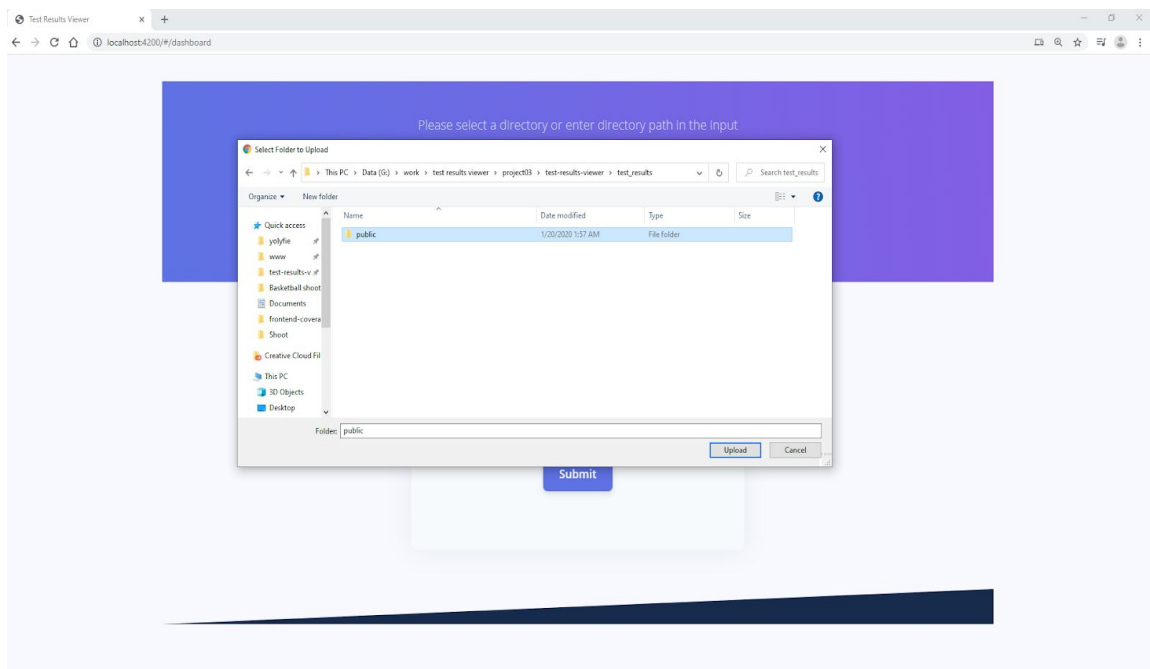
### 3. Using the project:

There are two options to select test data

Click on the select  button



Traverse the public folder you created previously and select it. then click on the upload button.

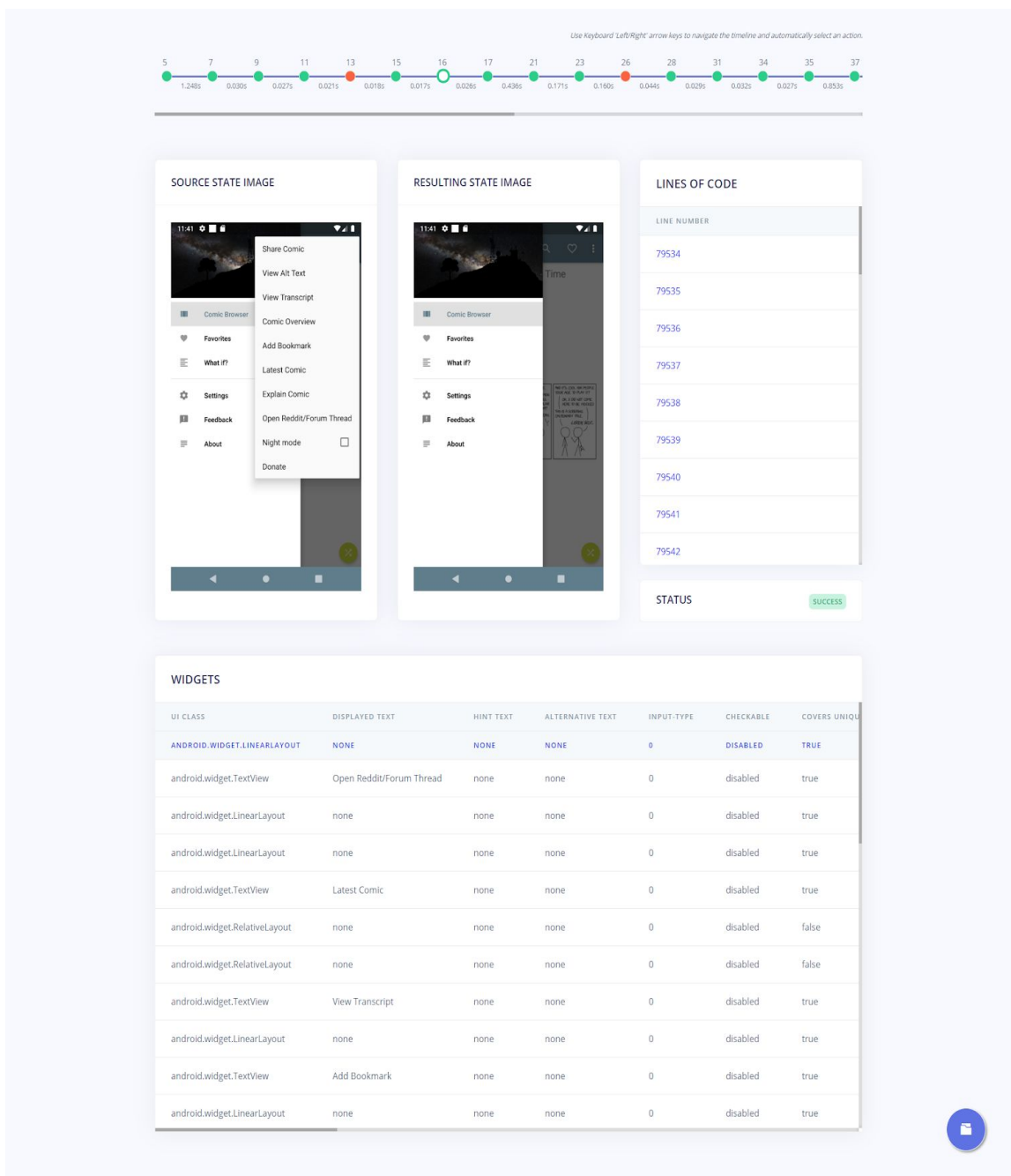


Or you can also enter the complete directory path in the input field.  
E.g: G:\work\test results viewer\project03\test-results-viewer\test\_results\public

The screenshot shows a web interface with a purple header and a light blue background. A central white modal box contains the following elements:

- Text: "Please select a directory or enter directory path in the input field to visualize the test results."
- Section: "Select" with a "Folder" button (represented by a folder icon).
- Text: "or enter folder path" with a red arrow pointing to the input field.
- Input field: Contains the text "G:\work\test results viewer\project03\test-res".
- Button: "Submit".

Finally, Click on the **Submit** button and you will be redirected to test results viewer dashboard.



#### **4. Code scaffolding:**

Run ``ng generate component component-name`` to generate a new component. You can also use ``ng generate directive|pipe|service|class|guard|interface|enum|module``.

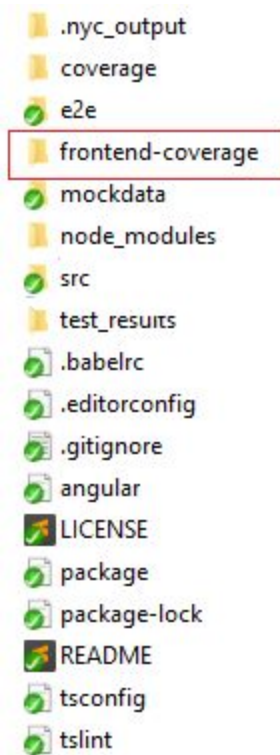
## 5. Running unit tests cases:

Run '`npm run frontend-test`' command to execute frontend test cases

Run '`npm run backend-test`' command to execute backend test cases

- Running unit tests with code coverage:

Run '`npm run frontend-test-coverage`' command, which will execute frontend unit test cases and will generate complete code coverage report in '`frontend-coverage`' folder.



Open index.html file from `frontend-coverage` folder to see the report in the browser.

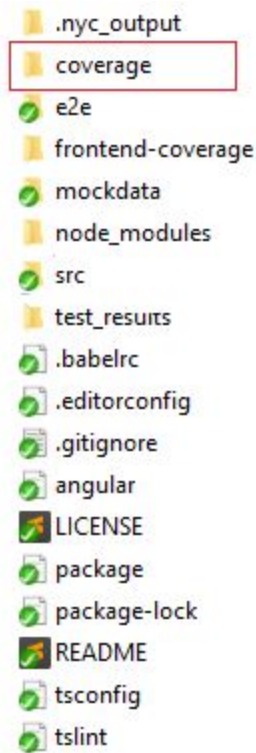
Sample report for the frontend code coverage:

All files  
90.27% Statements (33/237) 77.03% Branches (33/74) 87.04% Functions (47/54) 89.96% Lines (33/237)  
Press n or / to go to the next uncovered block, b, p or k for the previous block.

File	Statements	Branches	Functions	Lines
src	100%	3/3	100%	3/3
src/app	100%	18/18	100%	16/16
src/app/components/timeline	100%	68/68	100%	65/65
src/app/layouts/main-layout	100%	4/4	100%	3/3
src/app/pages/dashboard	84.76%	139/164	66%	128/152



Run '*npm run backend-test-coverage*', which will execute backend test cases and will generate complete code coverage report in '*coverage*' folder(As shown in the below screenshot) .



Open index.html file from '*coverage*' folder to see the report in the browser.

Sample report for the backend code coverage:

**All files**  
96.25% Statements (34/354) 81% Branches (42/51) 91.97% Functions (32/35) 96.18% Lines (34/354)  
Press n or j to go to the next uncovered block, b, p or k for the previous block.

File	Statements	Branches	Functions	Lines
mockdata/json	100%	2/2	100%	0/0
src/backend	95.45%	21/22	50%	2/4
src/backend/controllers	100%	53/53	83.33%	10/12
src/backend/helpers	100%	27/27	93.75%	15/16
src/backend/routes	90.63%	29/32	50%	3/6
src/backend/services	95.73%	112/117	82.26%	51/62
src/backend/spec/controllers	92.06%	58/63	100%	0/0
src/backend/spec/helpers	93.48%	43/46	100%	0/0
src/backend/spec/routes	100%	55/55	100%	0/0
src/backend/spec/services	97.44%	114/117	100%	0/0

