

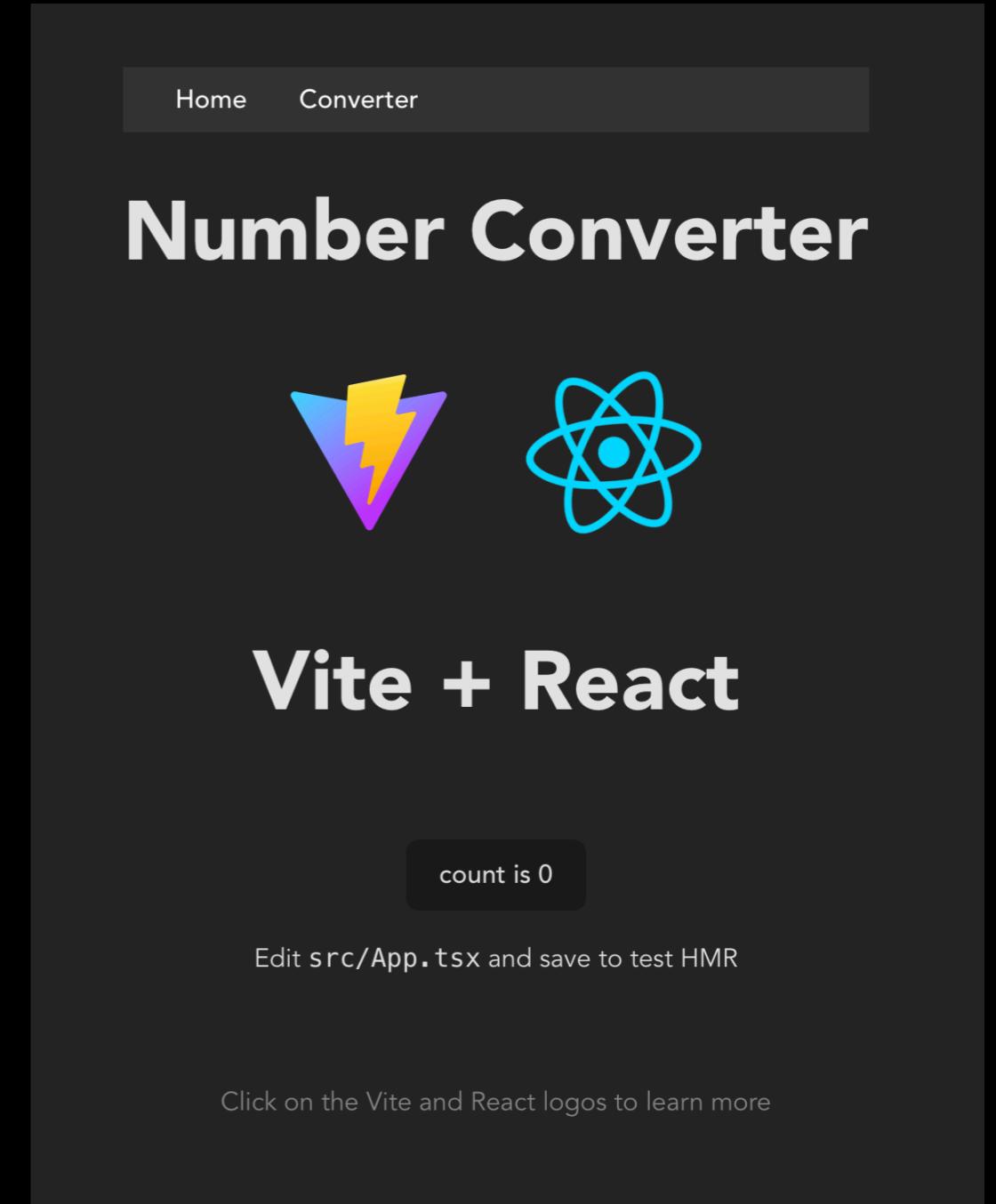
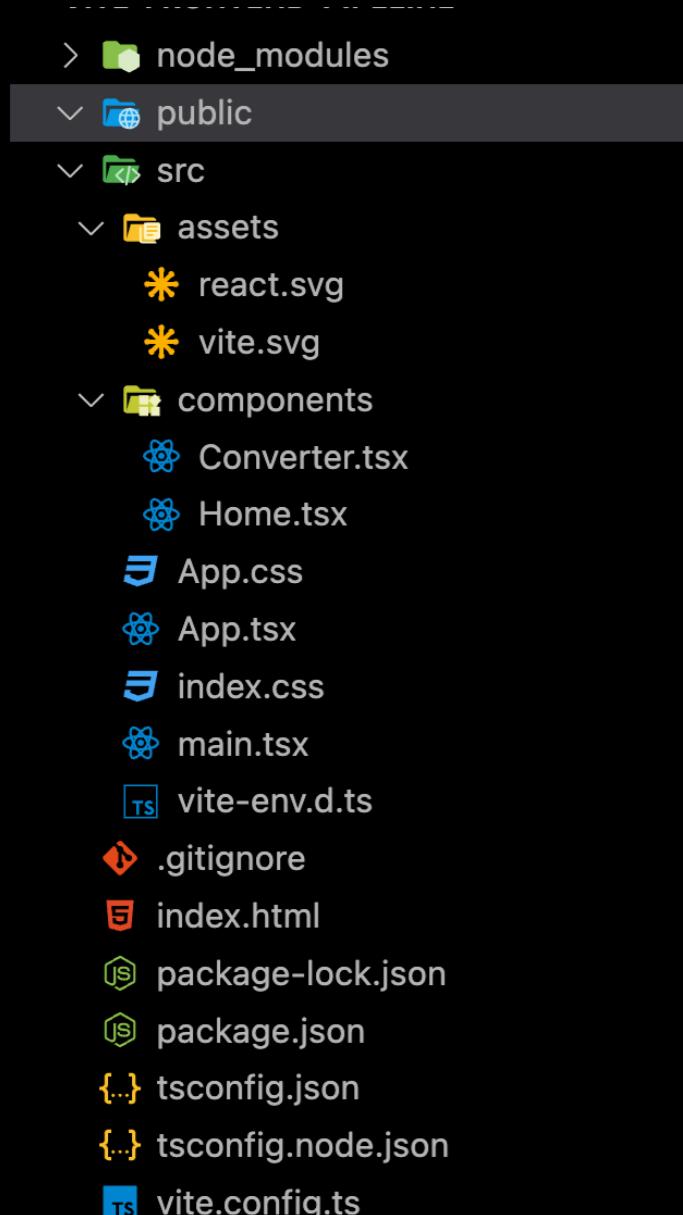
Frontend Pipeline

Quick guide to CI for frontend devs

Day 4 - Fundamentals of Frontend Development

Base Project

```
npm init vite@latest demo-project -- --template react-ts  
npm i react-router-dom
```



1. **Linting** and **formatting**: ESLint, Stylelint, Prettier
2. **Design system**: Storybook
3. **Unit testing**: Jest
4. **End-to-end** testing: Cypress
5. Workflow **automation**: CI with Github Actions, CD with Github Pages
6. Code **releases**

Linting Code

```
npm install eslint --save-dev  
npx eslint --init
```

```
apple ➜ ~/Dropbox/Software/vite-frontend-pipeline ➔ npx eslint --init  
You can also run this command directly using 'npm init @eslint/config'.  
✓ How would you like to use ESLint? · problems  
✓ What type of modules does your project use? · esm  
✓ Which framework does your project use? · react  
✓ Does your project use TypeScript? · No / Yes  
✓ Where does your code run? · browser  
✓ What format do you want your config file to be in? · JSON  
The config that you've selected requires the following dependencies:
```

```
eslint-plugin-react@latest @typescript-eslint/eslint-plugin@latest @typescript-eslint/parser@latest  
✓ Would you like to install them now? · No / Yes  
✓ Which package manager do you want to use? · npm  
Installing eslint-plugin-react@latest, @typescript-eslint/eslint-plugin@latest, @typescript-eslint/parser@latest
```

```
added 98 packages, and audited 282 packages in 12s
```

```
89 packages are looking for funding  
  run `npm fund` for details
```

```
found 0 vulnerabilities  
Successfully created .eslintrc.json file in /Users/natalie_agus/Dropbox/Software/vite-frontend-pipeline  
apple ➜ ~/Dropbox/Software/vite-frontend-pipeline ➔ █
```

Linting Code

```
npm install eslint-plugin-react-hooks --save-dev
```

```
{  
  "env": {  
    "browser": true,  
    "es2021": true  
  },  
  "extends": [  
    "eslint:recommended",  
    "plugin:react/recommended",  
    "plugin:react-hooks/recommended",  
    "plugin:react/jsx-runtime",  
    "plugin:@typescript-eslint/recommended",  
    "plugin:@typescript-eslint/recommended-requiring-type-checking"  
  ],  
  "overrides": [],  
  "parser": "@typescript-eslint/parser",  
  "parserOptions": {  
    "ecmaFeatures": {  
      "jsx": true  
    },  
    "ecmaVersion": 2021,  
    "sourceType": "module",  
    "project": "./tsconfig.json"  
  },  
  "plugins": ["react", "@typescript-eslint"],  
  "rules": {},  
  "settings": {  
    "react": {  
      "version": "detect"  
    }  
  }  
}
```

eslintrc.json

Linting Code

Ignore unnecessary files

The screenshot shows the VS Code interface with the Explorer sidebar open. The sidebar lists several files and folders under the 'VITE-FRONTEND-PIPELINE' project:

- .eslintignore (selected)
- .eslintrc.json
- .gitignore
- index.html
- package-lock.json
- package.json
- tsconfig.json
- tsconfig.node.json
- vite.config.ts

A second .eslintignore file is shown in the main editor area, containing the following content:

```
node_modules
dist
```

Linting Code

Add lint script in package.json

```
6  "scripts": {  
7    "dev": "vite",  
8    "build": "tsc && vite build",  
9    "preview": "vite preview",  
10   "lint:scripts": "eslint --ext .ts,.tsx src"  
11 },  
12 ...
```

```
apple ➜ ~/Dr/Software/vite-frontend-pipeline ➔ on git 🍭 main !3 ➔ npm run lint:scripts  
> vite-frontend-pipeline@0.0.0 lint:scripts  
> eslint --ext .ts,.tsx src
```

```
/Users/natalie_agus/Dropbox/Software/vite-frontend-pipeline/src/App.tsx  
25:9  error  Using target="_blank" without rel="noreferrer" (which implies rel="noopener") is a security risk in older browser  
  eners/#recommendations  react/jsx-no-target-blank  
28:9  error  Using target="_blank" without rel="noreferrer" (which implies rel="noopener") is a security risk in older browser  
  eners/#recommendations  react/jsx-no-target-blank  
  
/Users/natalie_agus/Dropbox/Software/vite-frontend-pipeline/src/components/Converter.tsx  
12:19  warning  'oldValue' is defined but never used  @typescript-eslint/no-unused-vars  
13:20  warning  'oldOrigin' is defined but never used  @typescript-eslint/no-unused-vars  
19:20  warning  'oldOrigin' is defined but never used  @typescript-eslint/no-unused-vars  
  
✖ 5 problems (2 errors, 3 warnings)  
2 errors and 0 warnings potentially fixable with the `--fix` option.
```

```
apple ➜ ~/Dr/Software/vite-frontend-pipeline ➔ on git 🍭 main !3 ➔
```

Linting Code

Fix the **errors** and **warnings**

```
const updateValue : (baseName: string, base: nu... =
  (baseName: string, base: number) : (e: ChangeEvent<HTMLInputElement... =>
  (e: React.ChangeEvent<HTMLInputElement>) : void => {
    if (e.target.value === "") {
      setValue(value: undefined);
      setOrigin(value: undefined);
    } else {
      setValue(prev : number | undefined) : number | undefined => {
        const newValue : number = parseInt(e.target.value, base);
        return isNaN(newValue) ? prev : newValue;
      });
      setOrigin(baseName);
    }
  };
}

  <Route path="/converter" element={<Converter />} />
</Routes>
<a href="https://vitejs.dev" target="_blank" rel="noopener">
  <img src={viteLogo} className="logo" alt="Vite logo" />
</a>
<div>
  <a href="https://reactjs.org" target="_blank" rel="noopener">
    <img src={reactLogo} className="logo react" alt="React logo" />
  </a>
</div>
```

Linting Code

Optional: customise the rules

```
"rules": {  
  "react/jsx-no-target-blank": [  
    "warn",  
    {  
      "allowReferrer": false  
    }  
  ]  
},
```

```
apple ➤ ~/Dr/Software/vite-frontend-pipeline ➤ on git 🌐 main !5 ➤ npm run lint:scripts
```

```
> vite-frontend-pipeline@0.0.0 lint:scripts  
> eslint --ext .ts,.tsx src
```

```
/Users/natalie_agus/Dropbox/Software/vite-frontend-pipeline/src/App.tsx  
 25:9 warning  Using target="_blank" without rel="noreferrer" (which implies rel="noopener") is a  
opener/#recommendations  react/jsx-no-target-blank  
 28:9 warning  Using target="_blank" without rel="noreferrer" (which implies rel="noopener") is a  
opener/#recommendations  react/jsx-no-target-blank  
  
* 2 problems (0 errors, 2 warnings)  
 0 errors and 2 warnings potentially fixable with the `--fix` option.
```

```
apple ➤ ~/Dr/Software/vite-frontend-pipeline ➤ on git 🌐 main !4 ➤
```

Linting Styles

```
npm install stylelint stylelint-config-standard stylelint-config-recess-order --save-dev
```

EXPLORER

OPEN EDITORS

GROUP 1

- .stylelintrc.json

GROUP 2

- .stylelintrc.json

VITE-FRONTEND-PIPELINE

- node_modules
- public
- src
 - .eslintignore
 - .eslintrc.json
 - .gitignore
 - .stylelintrc.json
- index.html
- package-lock.json
- package.json
- tsconfig.json
- tsconfig.node.json
- vite.config.ts

.stylelintrc.json

```
1 {  
2 "extends": ["stylelint-config-standard", "stylelint-config-recess-order"]  
3 }  
4
```

.stylelintignore

```
1 node_modules  
2 dist
```

TERMINAL

```
on git ↵ main !2 touch .stylelintrc.json  
on git ↵ main !2 ?1 touch .stylelintignore  
on git ↵ main !2 ?2
```

Linting Styles

Add lint script in package.json

package.json M X

package.json > {} scripts > abc lint:styles

```
7 "dev": "vite",
8 "build": "tsc && vite build",
9 "preview": "vite preview",
10 "lint:scripts": "eslint --ext .ts,.tsx src",
11 "lint:styles": "stylelint \"src/**/*.{css}\""
12 },
13 "dependencies": {
14   "eslint-plugin-react-hooks": "^4.6.0",
15   "react": "^18.2.0",
16   "react-dom": "^18.2.0",
17   "react-router-dom": "^6.6.2"
18 },
19 "devDependencies": {
```

TERMINAL

```
▶ ~ ~/Dr/Software/vite-frontend-pipeline on git ↵ main !2 ?1 touch .stylelintignore
▶ ~ ~/Dr/Software/vite-frontend-pipeline on git ↵ main !2 ?2 npm run lint:styles

> vite-frontend-pipeline@0.0.0 lint:styles
> stylelint "src/**/*.{css}"
```

src/App.css	
4:3	✖ Expected "padding" to come before "margin"
13:1	✖ Expected empty line before rule
16:1	✖ Expected empty line before rule
24:3	✖ Expected empty line before rule
43:1	✖ Expected class selector ".App" to be kebab-case
47:1	✖ Expected class selector ".App" to be kebab-case
50:21	✖ Expected "#333333" to be "#333"
51:3	✖ Expected "color" to come before "background-color"
51:10	✖ Expected "#ffffff" to be "#fff"
52:3	✖ Expected "padding" to come before "color"
55:1	✖ Expected class selector ".App" to be kebab-case
56:10	✖ Expected "#ffffff" to be "#fff"
58:3	✖ Expected "margin-left" to come before "text-decoration"
59:3	✖ Expected "margin-right" to come before "margin-left"
62:1	✖ Expected class selector ".Converter" to be kebab-case
64:3	✖ Expected "margin" to come before "text-align"
67:1	✖ Expected class selector ".Converter" to be kebab-case
72:1	✖ Expected class selector ".Converter" to be kebab-case

```
src/index.css
5:3  ✕ Expected "font-weight" to come before "line-height"          order/properties-order
7:3  ✕ Unexpected empty line before declaration                   declaration-empty-line-
8:10 ✕ Expected modern color-function notation                 color-function-notation
8:30 ✕ Expected "0.87" to be "87%"                           alpha-value-notation
11:3 ✕ Unexpected empty line before declaration                declaration-empty-line-
```

Linting Styles

Configure VSCode to fix files when saving happens

The screenshot shows the VSCode interface with the following components:

- EXPLORER**: Shows the project structure with files like package.json, settings.json, .vscode, node_modules, public, src, .eslintignore, .eslintrc.json, .gitignore, .stylelintignore, .stylelintrc.json, index.html, package-lock.json, package.json, tsconfig.json, tsconfig.node.json, and vite.config.ts.
- OPEN EDITORS**: Shows package.json and settings.json. settings.json is currently open, displaying the following JSON configuration:

```
.vscode > {..} settings.json > ...
1  {
2    "editor.codeActionsOnSave": {
3      "source.fixAll": true
4    }
5  }
6
```

- VITE-FRONTEND-PIPELINE**: Shows .pytest_cache and .vscode folders.
- TERMINAL**: Shows the command line history with the following commands:

```
on git ↵ main !2 ?2 mkdir .vscode
on git ↵ main !2 ?2 touch .vscode/settings.json
on git ↵ main !2 ?2
```

Linting Styles

Try auto fix: `npm run lint:styles -- --fix`

```
apple ✘ ~/Dr/So/vite-frontend-pipeline ➜ on git 🌐 main !3 ?2 ➔ npm run lint:styles -- --fix
```

```
> vite-frontend-pipeline@0.0.0 lint:styles
> stylelint "src/**/*.{css}
```

```
src/App.css
46:1 ✗ Expected class selector ".App" to be kebab-case          selector-class-pattern
50:1 ✗ Expected class selector ".App" to be kebab-case          selector-class-pattern
58:1 ✗ Expected class selector ".App" to be kebab-case          selector-class-pattern
65:1 ✗ Expected class selector ".Converter" to be kebab-case selector-class-pattern
70:1 ✗ Expected class selector ".Converter" to be kebab-case selector-class-pattern
75:1 ✗ Expected class selector ".Converter" to be kebab-case selector-class-pattern
```

```
6 problems (6 errors, 0 warnings)
```

```
apple ✘ ~/Dr/So/vite-frontend-pipeline ➜ on git 🌐 main !4 ?2 ➔
```

✓

Linting Styles

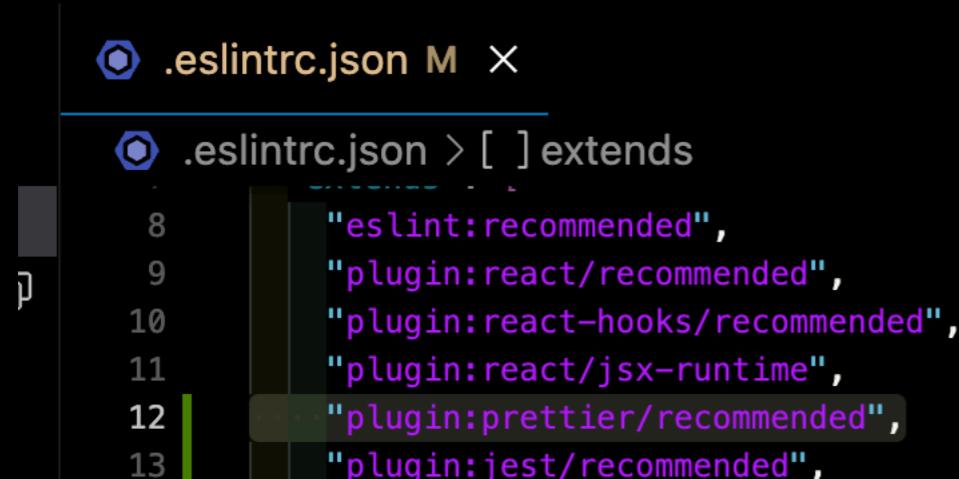
Disable the rule?

```
"rules": {  
  "selector-class-pattern": null  
}
```

Just be wise when you do this

Formatting

```
npm install prettier eslint-config-prettier eslint-plugin-prettier  
stylelint-config-prettier --save-dev
```



A screenshot of a code editor showing a file named `.eslintrc.json`. The file contains the following JSON configuration:

```
{  
  "extends": [  
    "eslint:recommended",  
    "plugin:react/recommended",  
    "plugin:react-hooks/recommended",  
    "plugin:react/jsx-runtime",  
    "plugin:prettier/recommended",  
    "plugin:jest/recommended"  
  ]  
}
```

- Update stylelintrc
- Update eslintrc
- Add prettierrc

Formatting

```
npm install prettier eslint-config-prettier eslint-plugin-prettier  
stylelint-config-prettier --save-dev
```

The screenshot shows the VS Code interface with two open files and a terminal window.

EXPLORER sidebar:

- OPEN EDITORS**:
 - GROUP 1**: .prettierrc (U)
 - GROUP 2**: .stylelintrc.json (M) - This file is selected.
- VITE-FRONTEND-PIPELINE**:
 - .pytest_cache
 - .vscode
 - node_modules
 - public
 - src** (●) - This folder is selected.
 - .eslintignore
 - .eslintrc.json
 - .gitignore
 - .prettierrc (U)
 - .stylelintignore
 - .stylelintrc.json (M)
 - index.html
 - package-lock.json
 - package.json
 - {...} tsconfig.json
 - {...} tsconfig.node.json
 - vite.config.ts

.prettierrc Editor:

```
1  {
2    "arrowParens": "avoid",
3    "bracketSpacing": true,
4    "htmlWhitespaceSensitivity": "css",
5    "insertPragma": false,
6    "jsxBracketSameLine": false,
7    "jsxSingleQuote": true,
8    "printWidth": 80,
9    "proseWrap": "always",
10   "quoteProps": "as-needed",
11   "requirePragma": false,
12   "semi": false,
13   "singleQuote": true,
14   "tabWidth": 2,
15   "trailingComma": "all",
16   "useTabs": false
17 }
```

.stylelintrc.json Editor:

```
1  {
2    "extends": [
3      "stylelint-config-standard",
4      "stylelint-config-recess-order",
5      "stylelint-config-prettier"
6    ],
7    "rules": {
8      "selector-class-pattern": null
9    }
10 }
```

TERMINAL:

```
apple ➤ ~/Dr/So/vite-frontend-pipeline ➤ on git ⌂ main !6 ?2 ➤ gaa; gcam "feat: add stylelint"  
[main c217027] feat: add stylelint  
8 files changed, 2047 insertions(+), 33 deletions(-)  
create mode 100644 .stylelintignore  
create mode 100644 .stylelintrc.json  
apple ➤ ~/Dropbox/Software/vite-frontend-pipeline ➤ on git ⌂ main ➤ npm install prettier eslint-config-  
added 6 packages, and audited 394 packages in 837ms  
105 packages are looking for funding  
  run `npm fund` for details  
found 0 vulnerabilities  
apple ➤ ~/Dr/Software/vite-frontend-pipeline ➤ on git ⌂ main !2 ➤ touch .prettierrc
```

Storybook

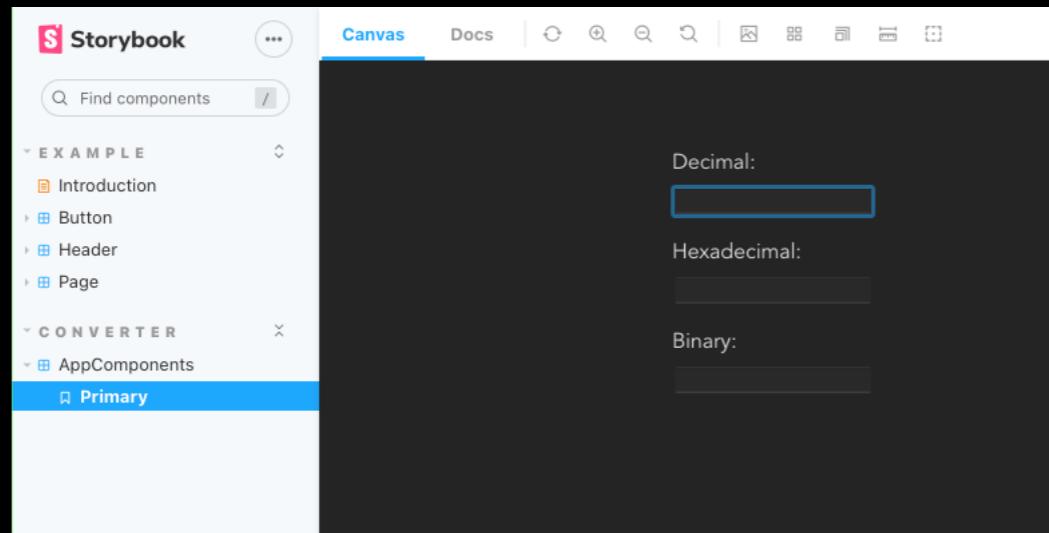
npx sb init --builder @storybook/builder-vite

The screenshot displays a developer's workspace with several open windows:

- EXPLORER**: Shows the project structure with files like `Converter.stories.tsx`, `Converter.tsx`, and `Home.tsx`.
- OPEN EDITORS**: Shows the code for `Converter.stories.tsx`. The code defines a story named "Primary" for the `Converter` component.
- VITE-FRONTEND-PIPELINE**: Shows the terminal output of the Vite build process, including the command `npm run storybook` and the resulting Storybook URL (`http://localhost:6006/`).
- TERMINAL**: Shows the command `npm run storybook` being run in the terminal.
- Storybook UI**: Shows the Storybook interface with the "Canvas" tab selected. It lists components like "Introduction", "Button", "Header", "Page", and "AppComponents". Under "AppComponents", the "Primary" story is highlighted.
- .eslintrc.json**: Shows the configuration for ESLint, specifically the "extends" section which includes `"plugin:storybook/recommended"`.

Storybook

Import custom css stylesheets in .storybook/preview.js

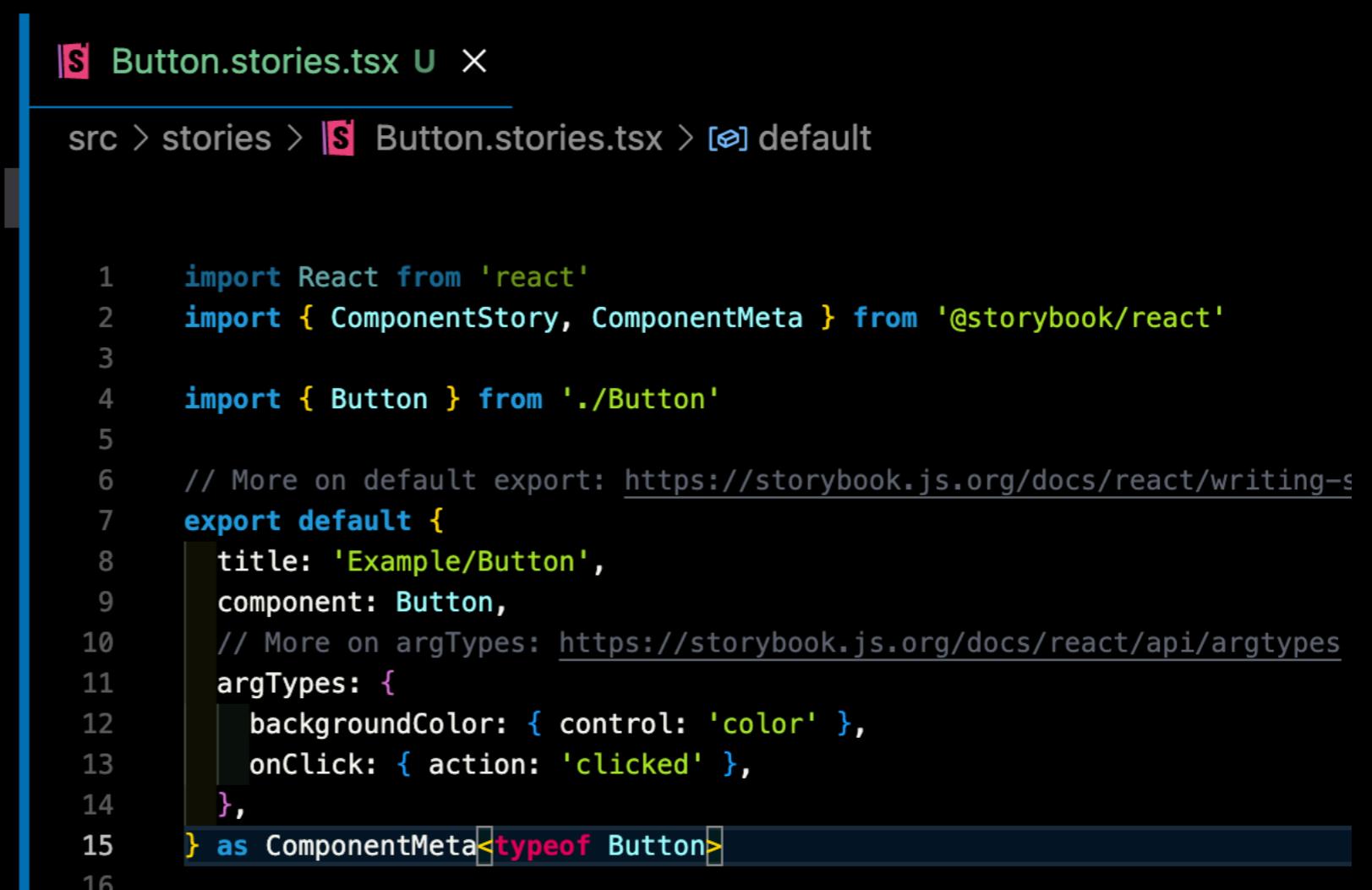


```
import '../src/index.css'
import '../src/App.css'
export const parameters = {
  actions: { argTypesRegex: '^on[A-Z].*$' },
  controls: {
    matchers: {
      color: /(background|color)$/i,
      date: /Date$/,
    },
  },
}
```

Storybook addon

```
npm install @storybook/addon-actions --save-dev
```

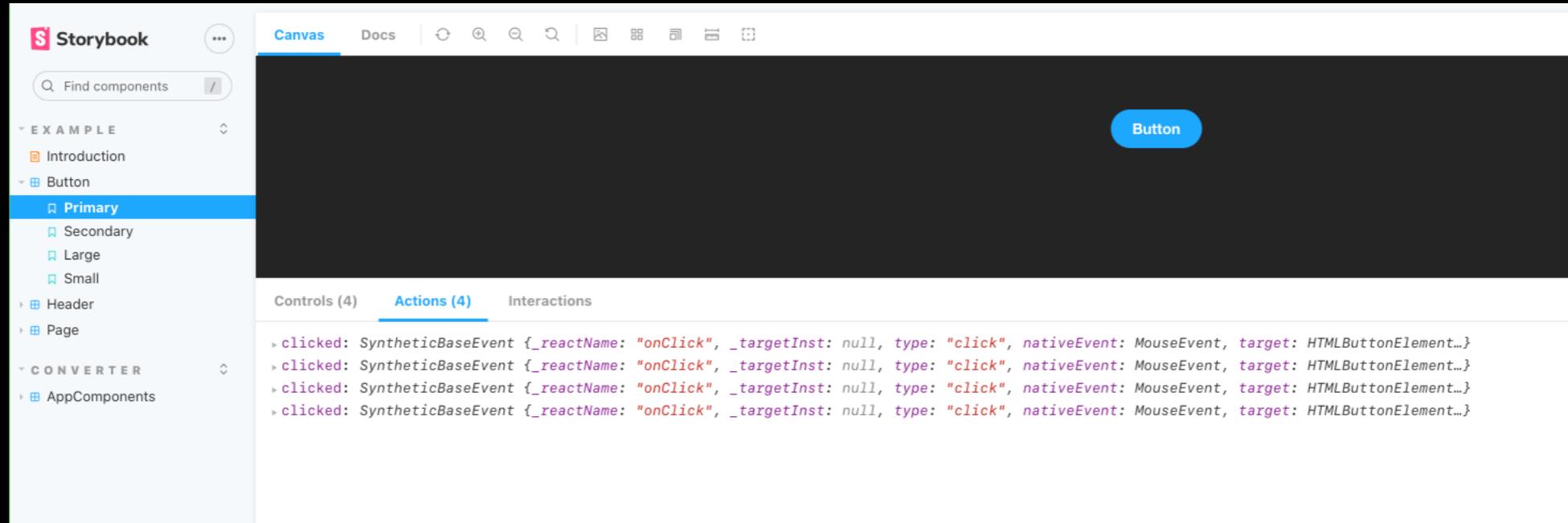
```
addons: [
  '@storybook/addon-links',
  '@storybook/addon-essentials',
  '@storybook/addon-actions',
  '@storybook/addon-interactions',
],
```



```
1 import React from 'react'
2 import { ComponentStory, ComponentMeta } from '@storybook/react'
3
4 import { Button } from './Button'
5
6 // More on default export: https://storybook.js.org/docs/react/writing-stories
7 export default {
8   title: 'Example/Button',
9   component: Button,
10  // More on argTypes: https://storybook.js.org/docs/react/api/argtypes
11  argTypes: {
12    backgroundColor: { control: 'color' },
13    onClick: { action: 'clicked' },
14  },
15} as ComponentMeta<typeof Button>
16
```

Storybook addon

npm install @storybook/addon-actions --save-dev



Controls (4)	Actions (4)	Interactions
Name		Control
primary		<input checked="" type="radio"/> False <input type="radio"/> True
label*		Button
backgroundColor		<input type="color"/> Choose color...
size		<input type="radio"/> small <input type="radio"/> medium <input type="radio"/> large
onClick		-

Unit Testing

```
npm install @types/jest @testing-library/react @testing-library/jest-dom  
jest ts-jest jest-environment-jsdom
```



```
package.json M X  
package.json > {} scripts  
  type: module,  
  ▷ Debug  
  "scripts": {  
    "dev": "vite",  
    "build": "tsc && vite build",  
    "preview": "vite preview",  
    "lint:scripts": "eslint --ext .ts,.tsx src",  
    "lint:styles": "stylelint \"src/**/*.{css,less}\"",  
    "storybook": "start-storybook -p 6006",  
    "build-storybook": "build-storybook",  
    "test": "jest"  
  }.
```

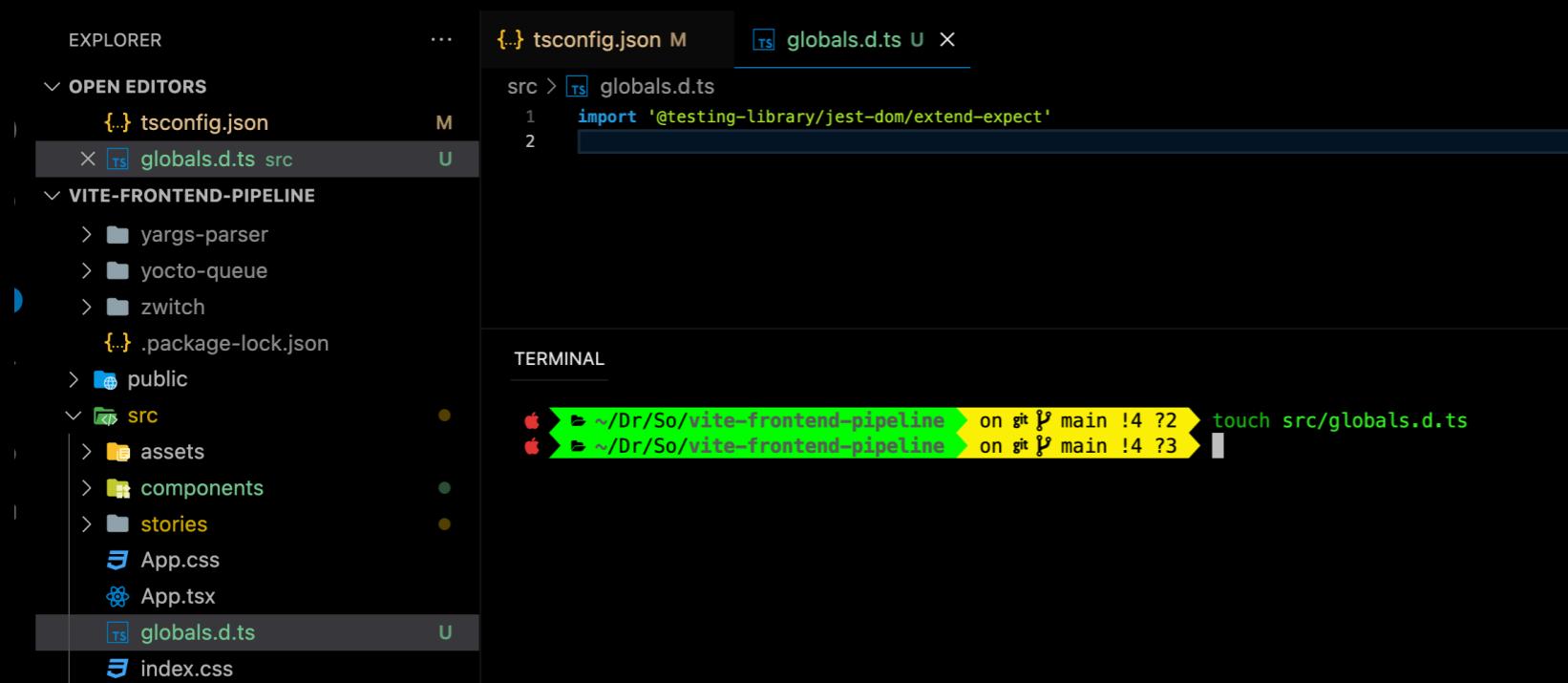
Create jest.config.js

```
export default {  
  
  preset: 'ts-jest',  
  // The root of your source code, typically /src  
  // `<rootDir>` is a token Jest substitutes  
  roots: ['<rootDir>/src'],  
  // Jest transformations -- this adds support for  
  // TypeScript  
  // using ts-jest  
  transform: {  
    '^.+\\.tsx?$': 'ts-jest',  
  },  
  preset: 'ts-jest',  
  
  testEnvironment: 'jest-environment-jsdom',  
  // Runs special logic, such as cleaning up  
  // components  
  // when using React Testing Library and adds special  
  // extended assertions to Jest  
  setupFilesAfterEnv: ['@testing-library/jest-dom/  
  extend-expect'],  
  
  // Test spec file resolution pattern  
  // Matches parent folder `__tests__` and filename  
  // should contain `test` or `spec`.  
  testRegex: '(/__tests__|.*|(\\.|/)(test|spec))\\.  
  tsx?$',  
  
  // Module file extensions for importing  
  moduleFileExtensions: ['ts', 'tsx', 'js', 'jsx',  
  'json', 'node'],  
}
```

Unit Testing

Setups: esModuleInterop

```
{...} tsconfig.json M X  
{...} tsconfig.json > {} compilerOptions  
1 {  
2   "compilerOptions": {  
3     "target": "ESNext",  
4     "useDefineForClassFields": true,  
5     "lib": ["DOM", "DOM.Iterable", "ESNext"],  
6     "allowJs": false,  
7     "skipLibCheck": true,  
8     "esModuleInterop": true,  
9     "allowSyntheticDefaultImports": true
```



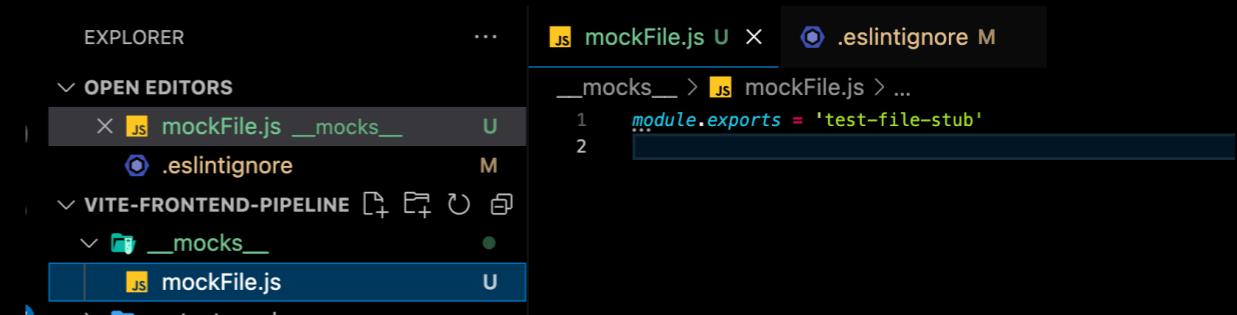
Also, see awesome-jest

Unit Testing

Mock static files: npm install identity-obj-proxy --save-dev

App.tsx

- It imports static files (*.svg)
- It utilises react router

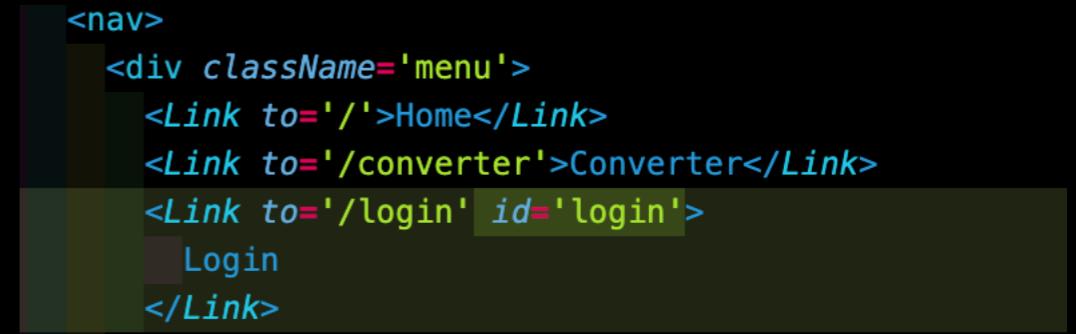


Add to jest.config.js

```
moduleNameMapper: {  
  '\\.(jpg|jpeg|png|gif|eot|otf|webp|svg|ttf|woff|woff2|mp4|webm|wav|mp3|m4a|aac|oga)$': '  
    <rootDir>/__mocks__/mockFile.js',  
  '\\.(css|less|scss|sass)$': 'identity-obj-proxy',  
},
```

Unit Testing

Testing with Router



```
import '@testing-library/jest-dom'
import { render, screen } from '@testing-library/react'
import App from '../App'
import { MemoryRouter } from 'react-router-dom'
import { act } from 'react-dom/test-utils'

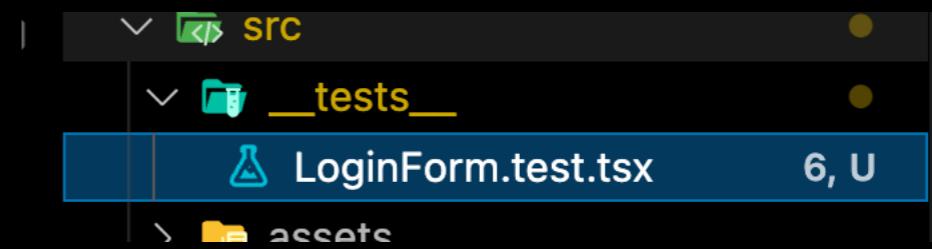
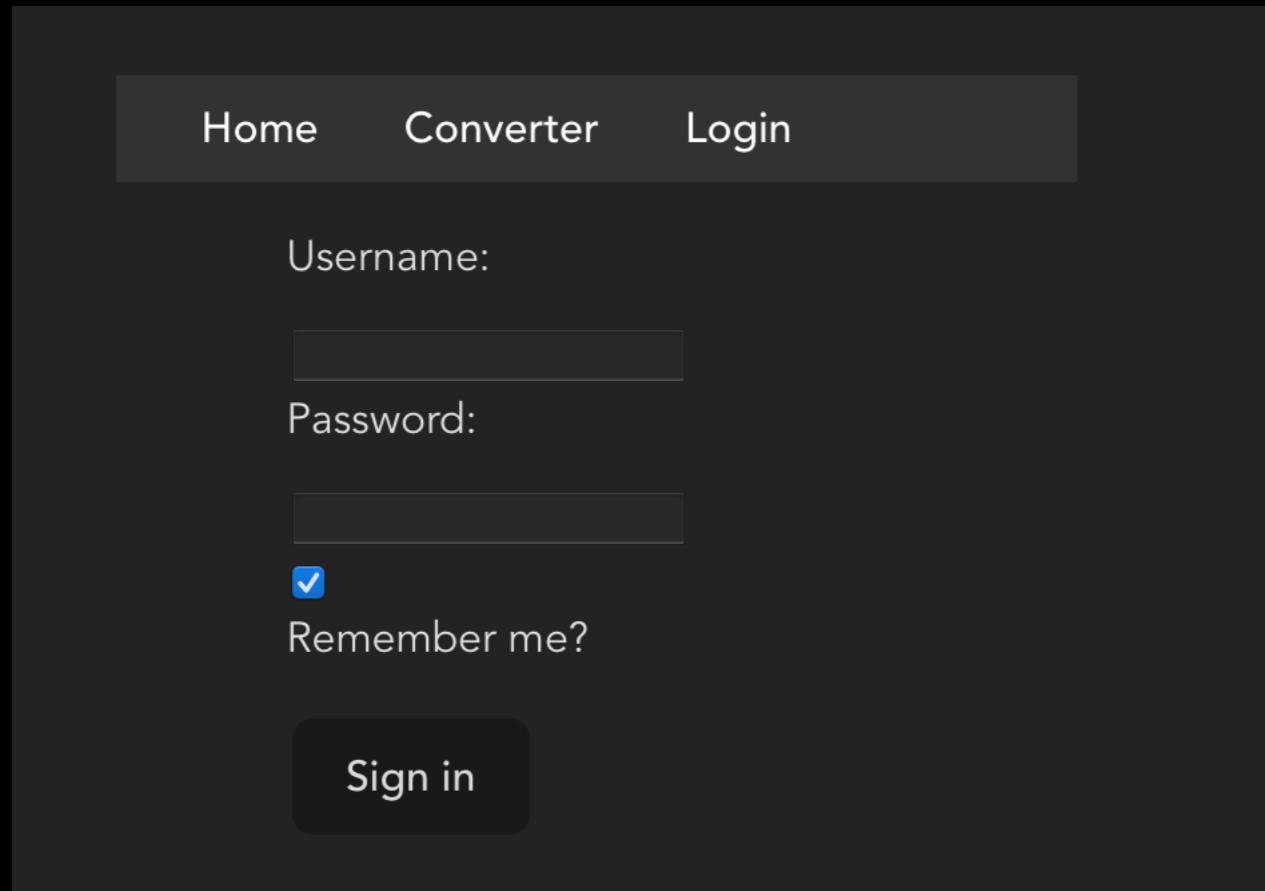
it('renders hello message', () => {
  // in a real test a renderer like "@testing-library/react"
  // would take care of setting up the DOM elements
  const root = document.createElement('div')
  document.body.appendChild(root)

  render(
    <MemoryRouter initialEntries={['/']}>
      <App />
    </MemoryRouter>,
  )
  expect(screen.getByText('Vite + React')).toBeInTheDocument()

  // Interact with, go to Login link
  act(() => {
    // Find the link (perhaps using the text content)
    const goLoginLink = document.querySelector('#login')
    // Click it
    goLoginLink?.dispatchEvent(new MouseEvent('click', { bubbles: true }))
  })
})
```

Test LoginForm Render

Setup a basic `LoginForm.tsx`



Test LoginForm

Event Handling

Ensure that the user can interact with our form in the **expected** ways:

- **Enter** a username
- **Enter** a password
- **Check** or **uncheck** "Remember me"
- **Submit** the form

Test LoginForm

Test hooks: npm install --save-dev @testing-library/react-hooks

```
1 import { useState, useCallback } from 'react'
2
3 interface UseCounterHookReturnType {
4   count: number
5   increment: () => void
6 }
7
8 function useCounter(): UseCounterHookReturnType {
9   const [count, setCount] = useState(initialState: 0)
10
11   const increment : () => void = useCallback(() : void => setCount(x : number => x + 1), [])
12
13   return { count, increment }
14 }
15
16 export default useCounter
17
```

```
src > __tests__ > 🚧 useCounter.test.tsx > ...
1 import { renderHook, act } from '@testing-library/react-hooks'
2 import useCounter from '../hooks/useCounter'
3
4 test(name: 'should increment counter', () : void => {
5   const { result } = renderHook(() : UseCounterHookReturnType => useCounter())
6
7   act(() : void => {
8     result.current.increment()
9   })
10
11   expect(result.current.count).toBe(expected: 1)
12 })
13
```

jest-cheat-sheet

eslint + jest

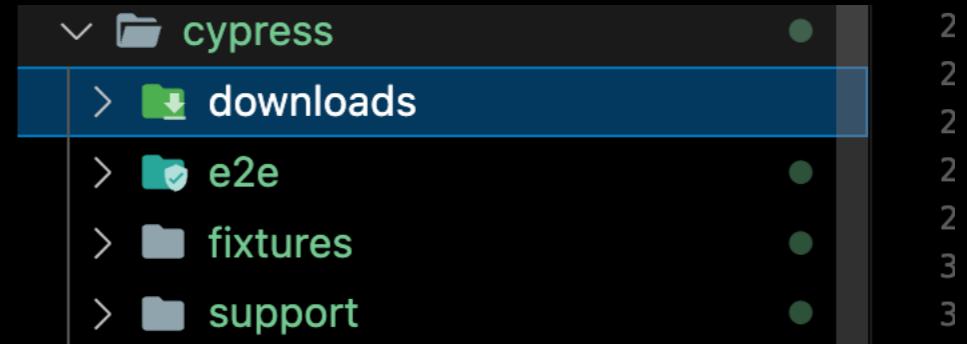
npm install eslint-plugin-jest --save-dev

```
❶ .eslintrc.json M X
❷ .eslintrc.json > {} env > ↗ jest
1  {
2    "env": {
3      "browser": true,
4      "es2021": true,
5      "jest": true
6    },
7    "extends": [
8      "eslint:recommended",
9      "plugin:react/recommended",
10     "plugin:react-hooks/recommended",
11     "plugin:react/jsx-runtime",
12     "plugin:prettier/recommended",
13     "plugin:jest/recommended",
14     "plugin:@typescript-eslint/recommended",
15     "plugin:@typescript-eslint/recommended-requiring-type-checking"
16   ],
17   "overrides": [],
18   "parser": "@typescript-eslint/parser",
19   "parserOptions": {
20     "ecmaFeatures": {
21       "jsx": true
22     },
23     "ecmaVersion": 2021,
24     "sourceType": "module",
25     "project": "./tsconfig.json"
26   },
27   "plugins": ["react", "@typescript-eslint", "jest"],
28   "rules": {
29     "react/jsx-no-target-blank": [
30       "warn",
31     ]
32   }
33 }
```

End-to-end Testing

npm install cypress --save-dev

```
package.json > {} scripts
12   "storybook": "start-storybook -p 6006",
13   "build-storybook": "build-storybook",
14   "test": "jest",
15   "test:watch": "jest --watch",
16   "cypress:open": "cypress open"
17 },
```

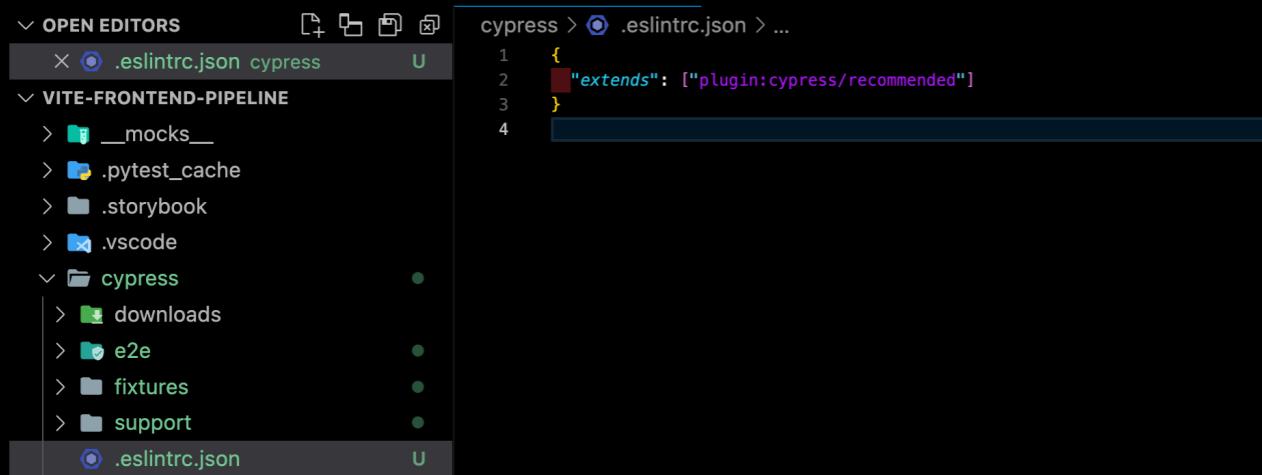


npm run cypress:open

A screenshot of the Cypress UI interface. On the left, there's a sidebar with navigation links: 'vite-frontend-pipeline' (selected), 'Specs', 'Runs', and 'Settings'. The main area is titled 'Specs' and shows a search bar with 'Search specs' and a result count of '20 matches'. A blue button labeled '+ New spec' is visible. Below the search bar, there's a section for 'E2E specs' with a tree view. Under 'cypress / e2e', there are two collapsed sections: '1-getting-started' containing 'todo.cy.js' and '2-advanced-examples' containing 'actions.cy.js', 'aliasing.cy.js', 'assertions.cy.js', 'connectors.cy.js', 'cookies.cy.js', and 'cypress-apis.cy.js'. To the right of the tree view, there are three filters: 'Last updated', 'Latest runs', and 'Average duration'.

End-to-end Testing

```
npm install eslint-plugin-cypress @testing-library/cypress --save-dev
```

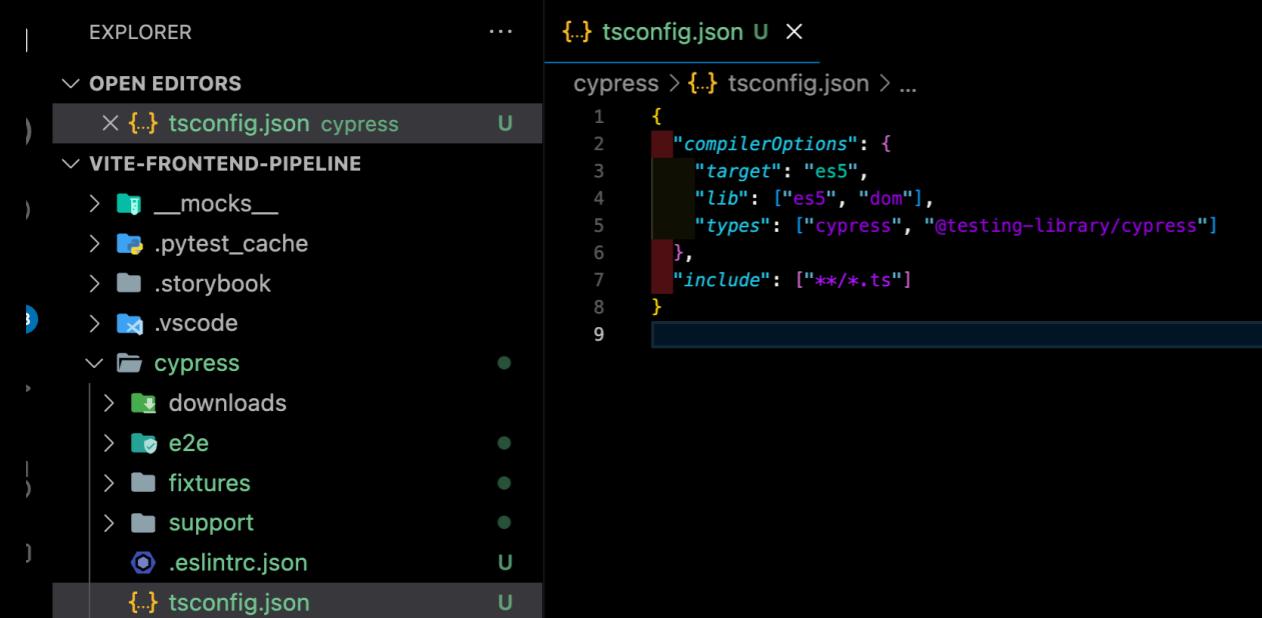


```
cypress > .eslintrc.json > ...
1 {
2   "extends": ["plugin:cypress/recommended"]
3 }
4
```

.eslintignore M X

.eslintignore

```
1 node_modules
2 dist
3 jest.config.js
4 vite.config.ts
5 __mocks__/
6 cypress/
7 cypress.config.ts
```



```
cypress > tsconfig.json > ...
1 {
2   "compilerOptions": {
3     "target": "es5",
4     "lib": ["es5", "dom"],
5     "types": ["cypress", "@testing-library/cypress"]
6   },
7   "include": ["**/*.ts"]
8 }
9
```

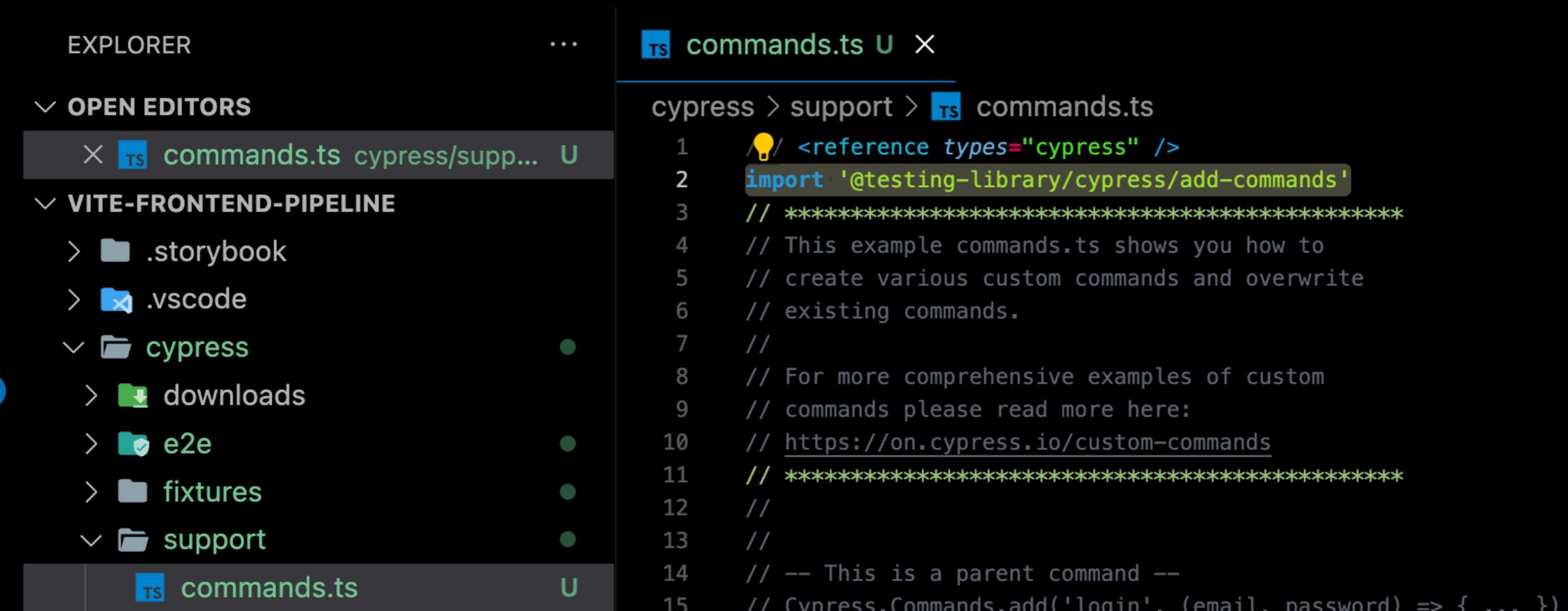
jest.config.js M X

jest.config.js > default

```
1 export default {
2   preset: 'ts-jest',
3   // The root of your source code, typically /src
4   // `<rootDir>` is a token Jest substitutes
5   roots: ['<rootDir>/src'],
6   // Jest transformations -- this adds support for TypeScript
7   // using ts-jest
8   transform: {
9     '^.+\\.tsx?$': 'ts-jest',
10   },
11
12   moduleNameMapper: {
13     '\\.(jpg|jpeg|png|gif|eot|otf|webp|svg|ttf|woff|woff2|mp4|webm|wav|mp3|m4a|aac|oga)$':
14       '<rootDir>/__mocks__/mockFile.js',
15     '\\.(css|less|scss|sass)$': 'identity-obj-proxy',
16   },
17   testMatch: ['**/src/**/?(*.)+(spec|test).[jt]s?(x)'],
```

End-to-end Testing

```
npm install eslint-plugin-cypress @testing-library/cypress --save-dev
```



EXPLORER

OPEN EDITORS

- commands.ts cypress/support

VITE-FRONTEND-PIPELINE

- .storybook
- .vscode
- cypress
 - downloads
 - e2e
 - fixtures
 - support
 - commands.ts

...

commands.ts

```
cypress > support > commands.ts
1 // <reference types="cypress" />
2 import '@testing-library/cypress/add-commands'
3 // ****
4 // This example commands.ts shows you how to
5 // create various custom commands and overwrite
6 // existing commands.
7 //
8 // For more comprehensive examples of custom
9 // commands please read more here:
10 // https://on.cypress.io/custom-commands
11 // ****
12 //
13 //
14 // -- This is a parent command --
15 // Cypress.Commands.add('login', (email, password) => { ... })
```

End-to-end Testing

Set baseUrl

cypress.config.ts

```
import { defineConfig } from 'cypress'

export default defineConfig({
  e2e: {
    setupNodeEvents(on, config) {
      // implement node event listeners here
    },
    baseUrl: 'http://localhost:5173',
  },
})
```

The screenshot shows a code editor with two tabs: 'cypress.config.ts' and 'app.cy.ts'. The 'app.cy.ts' tab is active, displaying a test script. The test script is part of a larger suite, as indicated by the breadcrumb navigation at the top of the code editor.

```
cypress > e2e > 3-custom-tests > app.cy.ts > describe('App') callback > it('increments the counter') callback
```

```
>>> 1 < describe(title: 'App', () : void => {
>>> 2 <| it(title: 'increments the counter', () : void => {
>>> 3 | cy.visit(url: '/')
> 4 | cy.findByRole(id: 'button', { name: /count is/i }).click()
> 5 <| cy.findByRole(id: 'button', { name: /count is/i }).should(
> 6 |   chainers: 'contain',
> 7 |   value: 'count is 1',
> 8 |
> 9 | cy.findByRole(id: 'button', { name: /count is/i }).click()
>10 <| cy.findByRole(id: 'button', { name: /count is/i }).should(
>11 |   chainers: 'contain',
>12 |   value: 'count is 2'.
```

The code editor highlights the 'cy.visit' line with a yellow circle icon, indicating it is the current line of interest or has been selected.

app.cy.ts

Test: button click, go to other route, page load

Specs ✓ 1 ✘ -- ⚡ ⏪ ⏴

app.cy.ts 366ms

App

✓ increments the counter

TEST BODY

```
1 visit http://localhost:5173/
2 findByRole button, {name: /count is/i}
3 -click
4 findByRole button, {name: /count is/i}
5 -assert expected <button> to contain count is
1
6 findByRole button, {name: /count is/i}
7 -click
8 findByRole button, {name: /count is/i}
9 -assert expected <button> to contain count is
2
10 get #navbar
11 -contains Converter
12 -click
13 (new url) http://localhost:5173/converter
14 -contains Hexadecimal
```

http://localhost:5173/converter

Chrome 109

1000x660 (90%)

Decimal:

Hexadecimal:

Binary:

Vite + React

count is 2

Edit src/App.tsx and save to test HMR



Github Pages + Actions

Create a public repo, action script

The screenshot shows a GitHub repository in VS Code. The left sidebar displays the repository structure, including files like .nvmrc, test-and-deploy.yml, .github/workflows/test-and-deploy.yml, .gitignore, .eslintrc.json, .prettierrc, and .stylelintignore. The main editor area shows three files: .nvmrc, test-and-deploy.yml, and .github/workflows/test-and-deploy.yml. The .nvmrc file contains the line `v16.17.0`. The test-and-deploy.yml file defines a workflow named 'Test and Deploy' that runs on push to the master branch. It includes steps for checkout, setup node, install dependencies, build project, run ESLint, run Stylelint, run tests using Cypress, and run Vite. The .github/workflows/test-and-deploy.yml file is also visible. The terminal at the bottom shows commands being run in the repository directory.

```
.nvmrc
v16.17.0

test-and-deploy.yml
name: Test and Deploy
on:
  push:
    branches:
      - master
jobs:
  build:
    name: Build
    runs-on: ubuntu-latest
  steps:
    - name: Checkout repo
      uses: actions/checkout@v2
    - name: Setup Node
      uses: actions/setup-node@v2
      with:
        node-version-file: '.nvmrc'
        cache: 'npm'
    - name: Install dependencies
      run: npm install
    - name: Build project
      run: npm run build
    - name: Run ESLint
      run: npm run lint:scripts
    - name: Run Stylelint
      run: npm run lint:styles
    - name: Run Tests
      run: npm run test-vite
    - name: Run Cypress
      uses: cypress-io/github-action@v5.0.0
      timeout-minutes: 10
      with:
        config-file: cypress.config.ts
        build: npm run build
        start: npm run preview -- --port=3000

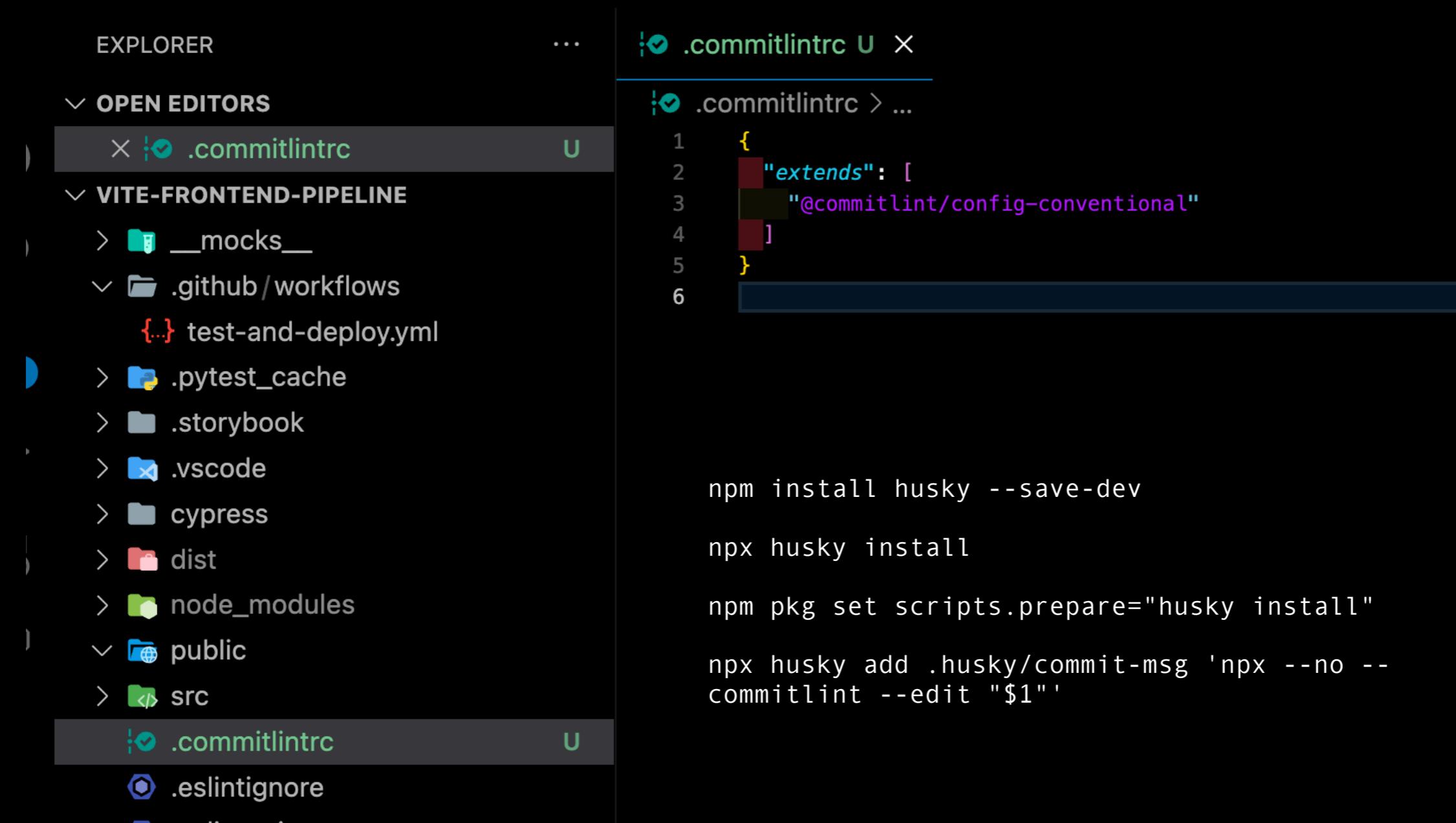
.github > workflows > test-and-deploy.yml
name: Test and Deploy
on:
  push:
    branches:
      - master
jobs:
  build:
    name: Build
    runs-on: ubuntu-latest
  steps:
    - name: Checkout repo
      uses: actions/checkout@v2
    - name: Setup Node
      uses: actions/setup-node@v2
      with:
        node-version-file: '.nvmrc'
        cache: 'npm'
    - name: Install dependencies
      run: npm install
    - name: Build project
      run: npm run build
    - name: Run ESLint
      run: npm run lint:scripts
    - name: Run Stylelint
      run: npm run lint:styles
    - name: Run Tests
      run: npm run test-vite
    - name: Run Cypress
      uses: cypress-io/github-action@v5.0.0
      timeout-minutes: 10
      with:
        config-file: cypress.config.ts
        build: npm run build
        start: npm run preview -- --port=3000
```

```
~/Dr/So/vite-frontend-pipeline on master mkdir .github
~/Dr/So/vite-frontend-pipeline on master mkdir .github/workflows
~/Dr/So/vite-frontend-pipeline on master touch .github/workflows/test-and-deploy.yml
~/Dr/So/vite-frontend-pipeline on master ?1 touch .nvmrc
~/Dr/So/vite-frontend-pipeline on master ?2
```

```
git remote add origin [REPO-URL]
git branch -M [master/main]
git push -u origin [master/main]
```

Commit Lint

```
npm install --save-dev @commitlint/config-conventional @commitlint/cli
```



EXPLORER

OPEN EDITORS

- ✓ .commitlintrc
- ✓ VITE-FRONTEND-PIPELINE
 - > __mocks__
 - ✓ .github/workflows
 - {...} test-and-deploy.yml
 - > .pytest_cache
 - > .storybook
 - > .vscode
 - > cypress
 - > dist
 - > node_modules
 - ✓ public
 - > src
 - .commitlintrc
 - .eslintignore

...

.commitlintrc **U** X

.commitlintrc > ...

```
1 {  
2 "extends": [  
3   "@commitlint/config-conventional"  
4 ]  
5 }  
6
```

npm install husky --save-dev
npx husky install
npm pkg set scripts.prepare="husky install"
npx husky add .husky/commit-msg 'npx --no --
commitlint --edit "\$1"'

Husky

Use commitlint seamlessly and set as git hook

```
apple ~ ~/Dr/So/vite-frontend-pipeline on 🐈 master !2 ?1 git add package.json
apple ~ ~/Dr/So/vite-frontend-pipeline on 🐈 master +1 !1 ?1 git commit -m "test bad commit"
✖ input: test bad commit
✖ subject may not be empty [subject-empty]
✖ type may not be empty [type-empty]

✖ found 2 problems, 0 warnings
ⓘ Get help: https://github.com/conventional-changelog/commitlint/#what-is-commitlint
```

husky - commit-msg hook exited with code 1 (error)

```
apple ~ ~/Dr/So/vite-frontend-pipeline on 🐈 master +1 !1 ?1 git commit -m "feat: add prepare script"
[master 26b68f4] feat: add prepare script
 1 file changed, 5 insertions(+), 1 deletion(-)
apple ~ ~/Dr/So/vite-frontend-pipeline on 🐈 master +1 !1 ?1
```

Semantic Release

```
npm install --save-dev semantic-release
```

```
{..} releaserc.json U X  
{..} releaserc.json > [ ] branches > abc 0  
1 {  
2   "branches": ["master"],  
3   "plugins": [  
4     "@semantic-release/commit-analyzer",  
5     "@semantic-release/release-notes-generator",  
6     "@semantic-release/github"  
7   ]  
8 }  
9
```

Semantic Release

Release workflow

```
{...} release.yml U X  
 .github > workflows > {...} release.yml  
 1   name: 'Release'  
 2   on:  
 3     push:  
 4       branches:  
 5         - master  
 6   jobs:  
 7     semantic-release:  
 8       runs-on: ubuntu-latest  
 9       steps:  
10       - uses: actions/checkout@v2  
11       - name: Setup Node.js  
12         uses: actions/setup-node@v2  
13         with:  
14           node-version-file: '.nvmrc'  
15           cache: 'npm'  
16       - name: Install dependencies  
17         run: npm install --legacy-peer-deps  
18       - name: Release a new version  
19         run: npx semantic-release  
20         env:  
21           github_token: '${{ secrets.GITHUB_TOKEN }}'  
22
```

Summary

1. **Linting** and **formatting**: ESLint, Stylelint, Prettier
2. **Design system**: Storybook
3. **Unit testing**: Jest
4. **End-to-end** testing: Cypress
5. Workflow **automation**: CI with Github Actions, CD with Github Pages
6. Code **releases**