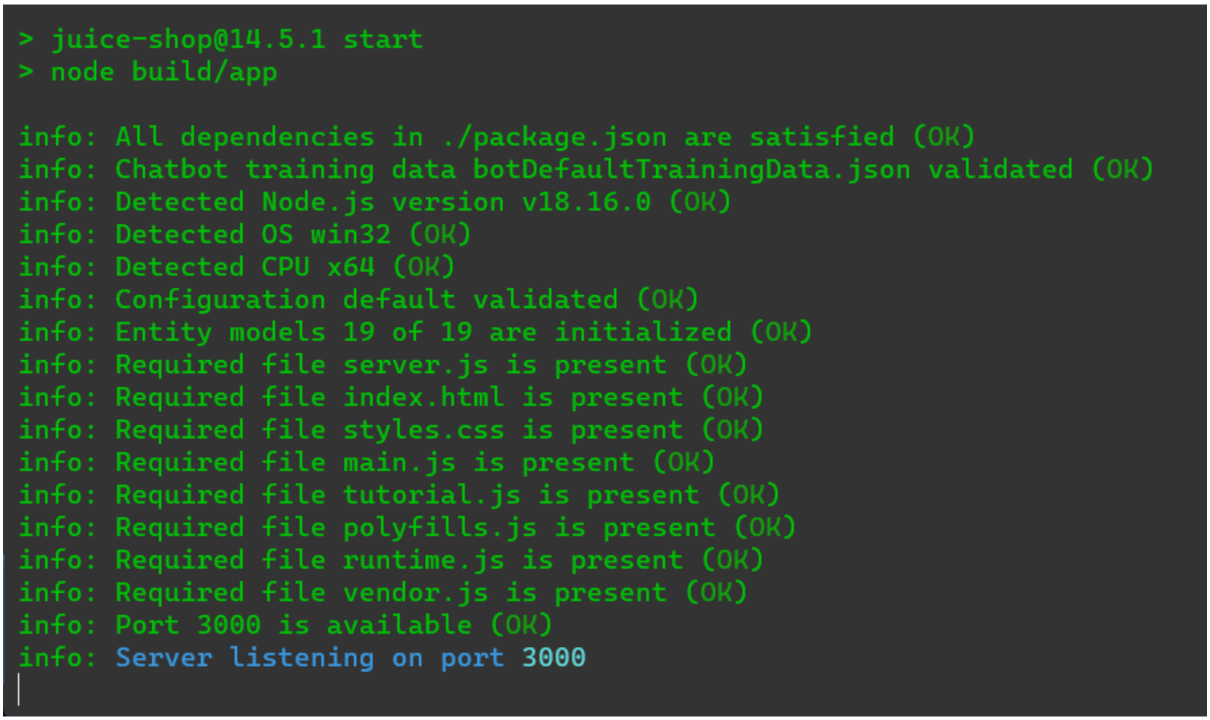
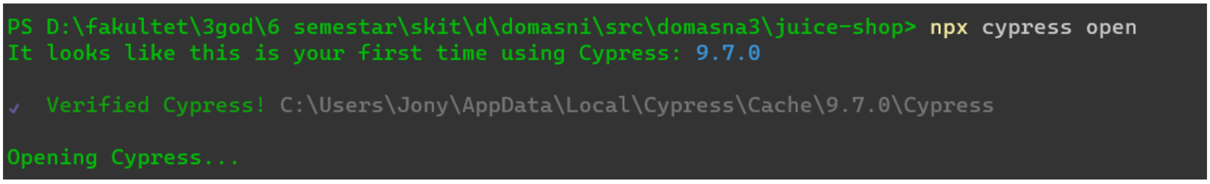
# СКИТ домашна 3

## Install locally juice shop app



```
> juice-shop@14.5.1 start
> node build/app

info: All dependencies in ./package.json are satisfied (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Detected Node.js version v18.16.0 (OK)
info: Detected OS win32 (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Server listening on port 3000
```

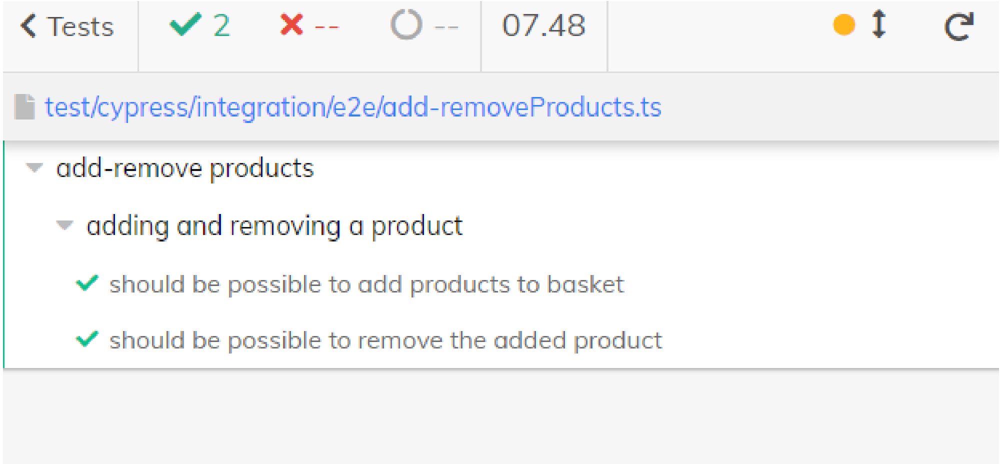


```
PS D:\fakultet\3god\6 semestar\skit\d\domasni\src\domasna3\juice-shop> npx cypress open
It looks like this is your first time using Cypress: 9.7.0

✓  Verified Cypress! C:\Users\Jony\AppData\Local\Cypress\Cache\9.7.0\Cypress

Opening Cypress...
```

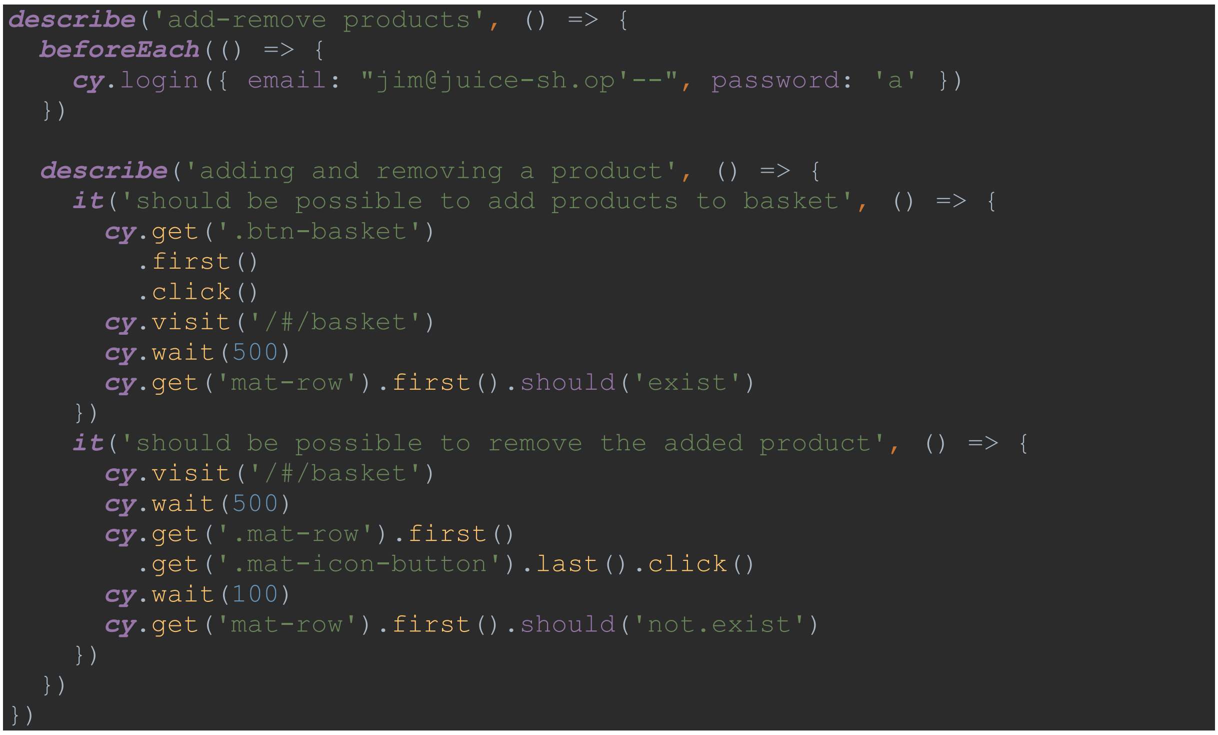
## Креирајте тест којшто ќе додава и ќе брише продукт



test/cypress/integration/e2e/add-removeProducts.ts

- add-remove products
  - adding and removing a product
    - ✓ should be possible to add products to basket
    - ✓ should be possible to remove the added product

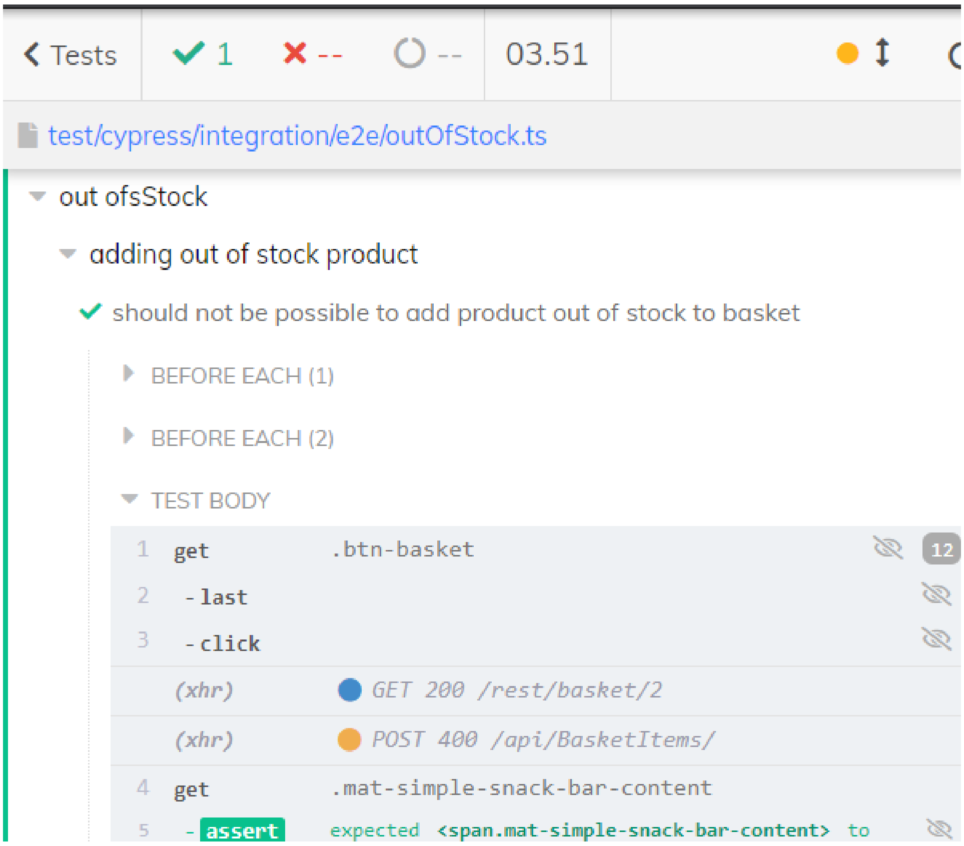
## add-removeProducts.ts



```
describe('add-remove products', () => {
  beforeEach(() => {
    cy.login({ email: "jim@juice-sh.op'--", password: 'a' })
  })

  describe('adding and removing a product', () => {
    it('should be possible to add products to basket', () => {
      cy.get('.btn-basket')
        .first()
        .click()
      cy.visit('/#/basket')
      cy.wait(500)
      cy.get('mat-row').first().should('exist')
    })
    it('should be possible to remove the added product', () => {
      cy.visit('/#/basket')
      cy.wait(500)
      cy.get('.mat-row').first()
        .get('.mat-icon-button').last().click()
      cy.wait(100)
      cy.get('mat-row').first().should('not.exist')
    })
  })
})
```

## Create test that will search item that is not in stock and validate that can’t be added to basket



test/cypress/integration/e2e/outOfStock.ts

- out ofsStock
  - adding out of stock product
    - ✓ should not be possible to add product out of stock to basket
      - BEFORE EACH (1)
      - BEFORE EACH (2)
      - TEST BODY

```
1  get        .btn-basket                                   12
2   - last
3   - click
   (xhr)      ● GET 200 /rest/basket/2
   (xhr)      ● POST 400 /api/BasketItems/
4  get        .mat-simple-snack-bar-content
5   - assert  expected <span.mat-simple-snack-bar-content> to
```

## outOfStock.ts

```
describe('out ofsStock', () => {
  beforeEach(() => {
    cy.login({ email: "jim@juice-sh.op'--", password: 'a' })
  })

  describe('adding out of stock product', () => {
    it('should not be possible to add product out of stock to basket', ()
=> {
      cy.get('.btn-basket')
        .last()
        .click()
      cy.get('.mat-simple-snack-bar-content')
        .should('have.text', 'We are out of stock! Sorry for the
inconvenience.')
      cy.visit('/#/basket')
      cy.wait(500)
      cy.get('mat-row').first().should('not.exist')
    })
  })
})
```