# Preconditions

Use the web application that you refactored in the previous home task.

# Task description

The goal of this task is to cover the web part of application with e2e tests. You can use testing tool of your choice, e.g. Cucumber.js, Protractor + Jasmine or NightWatch.js.

\*The home task should be implemented in Typescript.

\*It’s enough to use the e2e tests for the Chrome only.

# Evaluation Criteria

For every day of lateness there is a penalty in 0.5 point.

Maximum 10 points.

1. Application behavior should remain the same as it was before testing. [mandatory]
2. The package.json scripts sections contains a script test:e2e that runs E2E tests. [1 point]
3. E2E tests located in ./tests folder [1 point]
4. E2E tests cover the following cases [8 points]
   1. Password validation.
      1. Validate that submit button is disabled and the password again field has red boarders if the passwords do not match.
      2. Validate that submit button is enabled and the password again field has green boarders if the passwords match.
      3. Make sure that correct login is entered.
   2. Login availability.
      1. Mock the backend with your own minimal implementation.
      2. Verify that front-end renders red boarders for login input and disables the button, if the back-end returns status equals true.
      3. Verify that front-end renders green boarders for login input if the back-end returns status equal false.
      4. Make sure passwords are entered and they are match.
   3. Form submission.
      1. Mock the backend with your own minimal implementation.
      2. Verify that front-end sends POST /createAccount call to the BE and sends correct form data.
      3. Verify that in case BE returns status equal ok, front-end calls alert function with message ‘Account Created Successfully’.