T0511-1 - Front End Frameworks - (NOV20... MJ

Assignment 3: Everyday Market App - Routing and Form Validation (Graded)

Listen

Instructions
Weight: 25%

Type: Individual

Purpose: This assignment evaluates the following learning outcomes:

- Demonstrate how to implement routing with the Angular Router
- Demonstrate how to validate forms using client side validation in Angular

Instructions:

For this Assignment you will implement routing and develop a form with validation based on the Everyday Market App using Angular you created in Assignment 1.

You will not be able to follow Chapter 11 of the textbook step-by-step directly, but you should refer to it while completing this assignment. You should also make use of the Angular docs and any other useful online documentation (e.g. MDN Web Docs) to help you along, so that you can do the following:

- **Step 1:**

  - Analyze and revisit your Assignment 1. You may want to copy it over to a new folder. To do so, copy the `public` and `src` folders, as well as all the files below (such as package.json), then navigate to that folder and reinstall node_modules.
- **Step 2:**
  - Create a component RegisterPageComponent in CoreModule.
    - Users will be able to fill the form in with their details.
- **Step 3:**
  - Set up routes:
    - In CoreModule, set up routing so that '/register' shows the user the RegisterPageComponent.
    - In MarketModule, set up routing so that '/products' shows the user the ProductsPageComponent.
    - In app.routing.module.ts, set up a route so that attempting to go to '/' redirects the user to '/register'.
- **Step 4:**
  - Create a template-driven form with validation on RegisterPageComponent with the following fields. The fields should be of the right type to be validated as follows:
    - Name
      - Text input: only letters and space, at least 5 characters
    - Email address:
      - Text input: in valid email format
    - Phone number
      - Text input: 10 digit number
    - Date of birth:
      - Date picker: yyyy-mm-dd
    - Street address:
      - Text input: letters, numbers, or spaces
    - Province:
      - Dropdown: one of the Canadian provinces/territories
    - Country:

            - Dropdown: with United States and Canada as options, but must select Canada to proceed
        - Checkbox asking users to accept the Terms & Conditions (must be selected)
    - Beneath the form, place a Submit button which is disabled.
- **Step 5**:
    - Create a folder src/app/modules/core/register-page/validators
    - In this folder, create custom validators for your template-driven form to cover any validation cases that are not already provided by Angular. Apply these custom validators to the right form elements.
        - Hint: follow https://angular.io/guide/form-validation#adding-custom-validators-to-template-driven-forms
        - Hint: if you make a standalone validator directive, you will need to put it in the `imports` array of CoreModule. If you make a non-standalone directive, put it in the `declarations` array instead.
        - Hint: if you find a way to make a reusable validator which can handle more than one input, go for it!
        - It is all right to copy a regular expression from the internet, as long as you cite your sources using a comment above the copied line.
    - Show a visual indicator (e.g. red border) on fields that are `touched` AND `invalid`
- **Step 6:**
    - Once the user has filled in all the details in the correct format, the Submit button should become enabled.
    - Inject Angular's Router into the component and create an onSubmit() function in RegisterPageComponent. onSubmit() should use the injected Router to navigate to "/products".
    - Use (ngSubmit) in the form template to call onSubmit().
    - Upon clicking the Submit button, redirect the user to ProductsPageComponent.

Submission Guidelines:

- **You may not use AI-generated code anywhere in this assignment.**
- Test your application prior to submission.
- Please upload a zip folder with all required files to the assignment dropbox in Brightspace.

Assessment Criteria: You will be assessed using the grading rubric.

References:

Dresher, T., Zuker, A., & Friedman, S. (2018). Hands-On Full-Stack Web Development with ASP.NET Core. Packt Publishing. https://bookshelf.vitalsource.com/books/9781788627757

Submissions

No submissions yet. Drag and drop to upload your assignment below.

Drop files here, or click below!

Upload Record Choose Existing

You can upload files up to a maximum of 2 GB.

Comments

Paragraph

Reflect in ePortfolio

Activity Details

Task: Submit to complete this assignment

Due December 7 at 11:59 PM

Assessment

Grading Rubric - Assignments