



Angular კურსის დეტალურის სილაბუსი

კურსის აღწერა

მოცემული კურსი მოიცავს ვებ-დეველოპმენტის, კერძოდ Front-end მხრის, ერთ-ერთი ყველაზე პოპულარული ფრეიმვორკის (Angular 8) ფუნდამენტურ და მნიშვნელოვან ნაწილებს. კურსის განმავლობაში სტუდენტები შეისწავლიან ამ ტექნოლოგიის იმ პრინციპებსა და შესაძლებლობებს, რომლებიც მათ დაეხმარება შექმნან თანამედროვე და კონკურენტ უნარიანი აპლიკაციები.

რას ვისწავლით ამ კურსზე?

Angular 8-ის კურსის მსვლელობისას ჩვენ პირველ რიგში ვისწავლით ტაიპსკრიპტს საფუძვლებს, და შემდგომში შევეუდგებით ფრეიმვორკის შესწავლას, ვისწავლით მისი მოქმედების პრინციპებს, როგორებიცაა Routing, Data-Binding, Cross Component Communication, Observables და ა.შ. მათი მეშვეობით, კურსის მსვლელობის პარალელურად, შევექმნით კომპლექსურ და თანამედროვე SPA აპლიკაციას.

Course Content

TypeScript - შესავალი.

მოდულის მიზანია სტუდენტებმა ისწავლონ ტაიპსკრიპტის ის დამახასიათებელი ნიშნები, რომლებიც აუცილებელია ანგულარის აპლიკაციის დეველოპმენტისთვის.

- რა არის ტაიპსკრიპტი
- ტაიპსკრიპტის ტიპები
- კლასები
- ინტერფეისები
- დეკორატორები

Angular შესავალი

სტუდენტები შეისწავლიან ანგულარის საფუძვლებს, რათა შექმნან უმარტივესი

ვებ-აპლიკაციები. მოცემული მოდულის მსვლელობისას დავიწყებთ პროექტზე მუშაობას, რომელშიც ყოველ კვირას ჩავაშენებთ ახალ ნასწავლ თემებს.

- საფუძვლები.
- კომპონენტები
- მოდულები
- Angular CLI
- დირექტივები.

Debugging

მოცემულ მოდულში სტუდენტები ისწავლიან დებაგინგის პროცესს, რომელიც კოდის წერის აუცილებელი და საჭირო კონცეფციაა, ამის ცოდნა დეველოპმენტს ბევრად სასიამოვნოსა და პროდუქტიულს ხდის.

- ერორ მესიჯები
- კოდის დებაგინგი ბრაუზერში.
- Augury

Databinding & Lifecycle Hooks.

კურსის ამ ნაწილში ვისწავლით კომპონენტების Databinding-ის საფუძვლებს ამასთან ერთად დავიწყებთ Lifecycle hook - ების გარჩევასა და შესწავლას, რომლებიც ანგულარის აპლიკაციების უდიდეს ნაწილს იკავებენ.

- აპლიკაციის დაყოფა კომპონენტებად
- Property and event binding.
- Local references to Templates.
- თემფლეითზე წვდომა და დოკუმენტის ობიექტის მოდელი.
- კომპონენტის სიცოცხლის ციკლები.

სერვისები

ამ მოდულში შევეხებით სერვისებს, რომლებიც ფრეიმვორკის გამოყენებისთვის ერთ-ერთი ყველაზე კომფორტული და მნიშვნელოვანი ნაწილია.

- სერვისების შექმნა
- პროექტის მენეჯმენტი სერვისების გამოყენებით.
- სერვისების გამოყენება Cross-Component კომუნიკაციებისთვის.

Routing

კურსის ერთ-ერთი ყველაზე მნიშვნელოვანი ნაწილი არის როუტინგის შესახებ, რომლის მეშვეობით მომხმარებლები ვებ აპლიკაციის სხვადასხვა გვერდებს მიმოიხილავენ. ამ ნაწილში ვისწავლით თუ როგორ გაუმართოთ აპლიკაციის ნავიგაცია.

- რატომ როუტერი?
- ნავიგაციის გამართვა.
- როუტერ ლინკებით ნავიგაცია.
- Nested როუტები.
- Guards.

Observables

ანგულარის თანამედროვე აპლიკაცია წარმოუდგენელია Observable-ების გარეშე. ამიტომ სტუდენტები ისწავლიან RxJS ბიბლიოთეკის observable-ს და მისი გამოყენების წესებს.

- ანალიზი.
- Core of Observables.
- Observable-ს შექმნა.
- Subject.
- Behavior Subject.
- Subject vs Behavior Subject.

ფორმებთან მუშაობა

ამ მოდულში სტუდენტები ისწავლიან ფორმებთან მუშაობას, ანგულარი საშუალებას გვაძლევს ფორმები ორნაირად ვმართოთ. რა თქმა უნდა, ვისწავლით ამ ორივე მეთოდს და ვისაუბრებთ ორივეს დადებითსა და უარყოფით მხარეებზე.

რატომ გვჭირდება ანგულარის დახმარება ფორმებთან მუშაობაში?

- Template Driven forms.
- Creating form
- Submitting form.
- Understanding form state.
- Validations.
- Validation Error messages
- Default values.
- Reactive Forms.
- Creating form.
- Syncing HTML and form.
- Submitting the form.
- Validations.
- Custom Validators.
- Error handling.
- Template Driven vs Reactive Forms.



Http Requests

პროექტის ფრონტის მხარის წარმოდგენა Http - ის გამოყენების გარეშე წარმოუდგენელია, ის გვაძლევს სერვერთან კომუნიკაციის საშუალებას და აპლიკაციას ბევრად უფრო დინამიურს ხდის. ანგულარი გვაძლევს საკუთარ მეთოდს Http Request - ის გასაკეთებლად, რომელსაც განვიხილავთ კურსის მოცემულ მოდულში.

- ანგულარის ურთიერთქმედება სერვერთან.
- Http request - ის სტრუქტურა.
- GET Request.
- POST Request.
- RxJS ოპერატორების გამოყენება მონაცემების დასამუშავებლად.
- მონაცემების გამოტანა.
- სერვისების გამოყენება Http Request - ების გაკეთებისთვის.
- DELETE Request.
- Error Handling.
- ჰედერების შექმნა.