HTMLis ja CSS-s Menüü tegemine

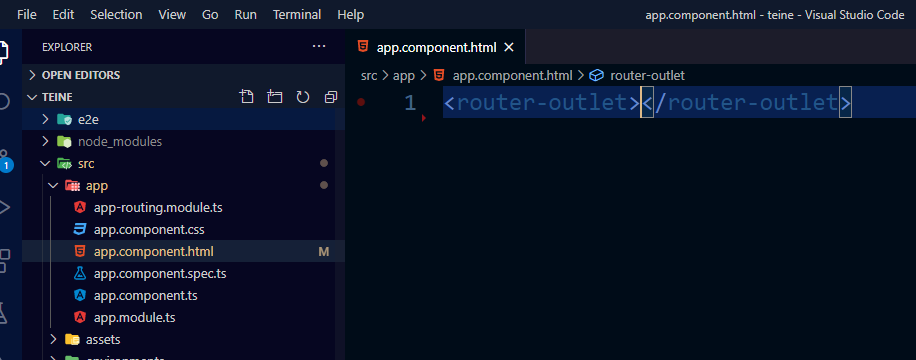
1. **Tee uus projekt**

Text

Description automatically generated

Routingule „yes“ pannes teeb Angular valmis vajalikud failid URL liikumiste jaoks – app-routing.module.ts ja loob ühendused

1. **Tee app.component.html tühjaks ja jäta vaid router-outlet**



Selle asemele tuleb seos, mis on kirjutatud app-routing.module.ts sees. Kui on localhost:4200/bla ja vaja näidata blaatens.component.html, siis on seos: { path: „bla“, component: BlaatensComponent }

1. **Genereeri uued componendid: navbar, kodu, uudised, kontakt, meist**

Selleks peab minema projekti kausta -> „cd teine“. (change directory)

Text

Description automatically generated

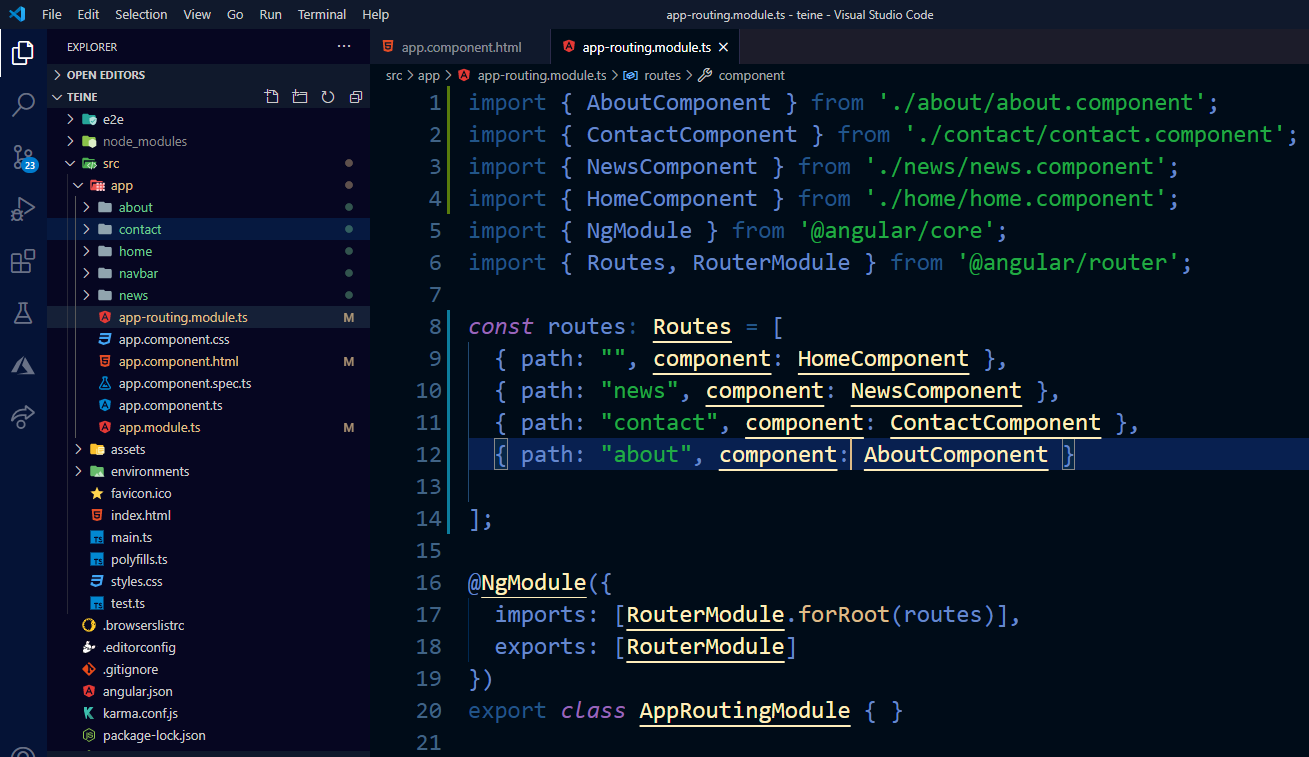
1. **Lisa app-routingusse, missugused URLid on olemas**

Localhost:4200 on kodu

Localhost:4200/news on uudised

Localhost:4200/contact on kontakt

Localhost:4200/about on meist



Pane tähele, et kõik componendid oleksid **imporditud**.

Pane tähele, et massiivi elemendid on eraldatud komadega:

[ <- massiivi (listi) algus

{}, <- massiivi (listi) element (eraldatud komaga)

{},

{} <- massiivi (listi) viimasele elemendile ei pea järgnema koma, aga VÕIB kui tahame, töötab

] <- massiivi (listi) lõpp

1. **Käivita „ng serve“ ja kontrolli**

Selleks peab olema „teine“ kaustas

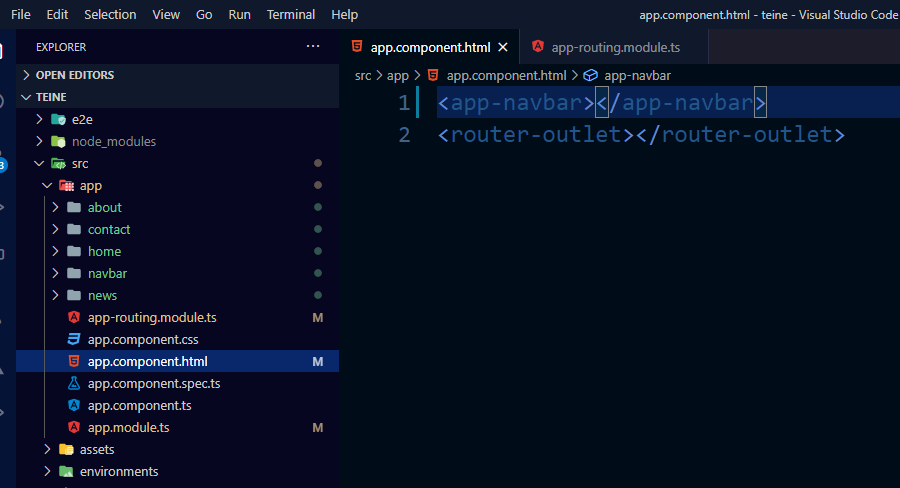
1. **Lisame app.component.html sisse app-navbar täägi**

App.component.html sees olevad koodiread on alati – igal leheküljel nähtavad.

Teised componendid on nähtavad kindlale URLile minnnes, see seos on loodud app-routingus.

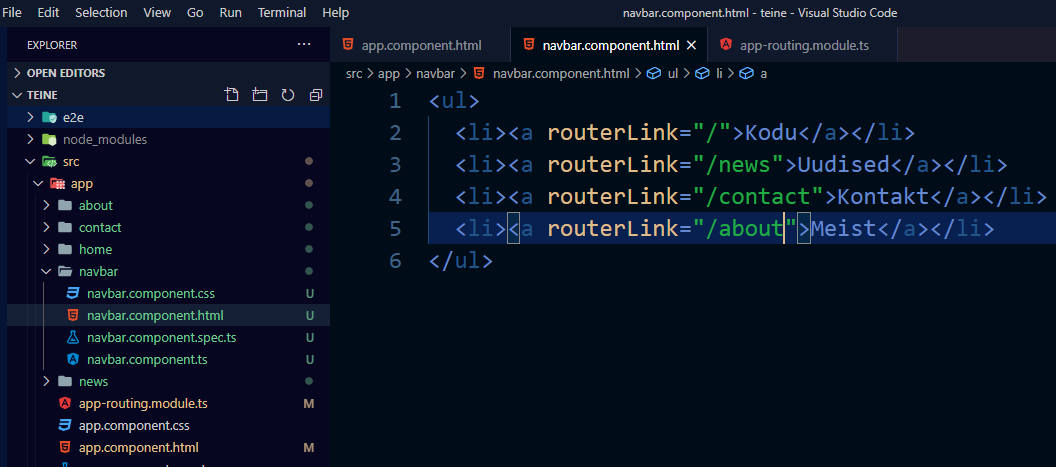
Navbar component on täägi (ingl k. tag. Tag on ka <img>, <div>, <p> jne) abil nüüd koguaeg nähtav (vaata alt).

Selle abil suuname navbar componenti siia nähtavaks, mitte kindla URL alusel.



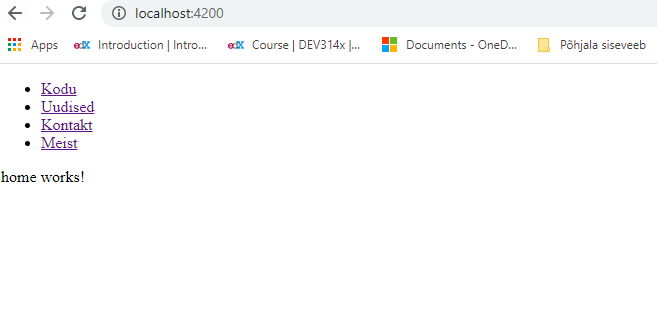
Mine lehele ja kontrolli, et see on koguaeg olemas (kirjuta aadressiribale ise käsitsi: localhost:4200, localhost:4200/news, localhost:4200/contact, localhost:4200/about)

1. **Täidame navbar componendi HTMLi**

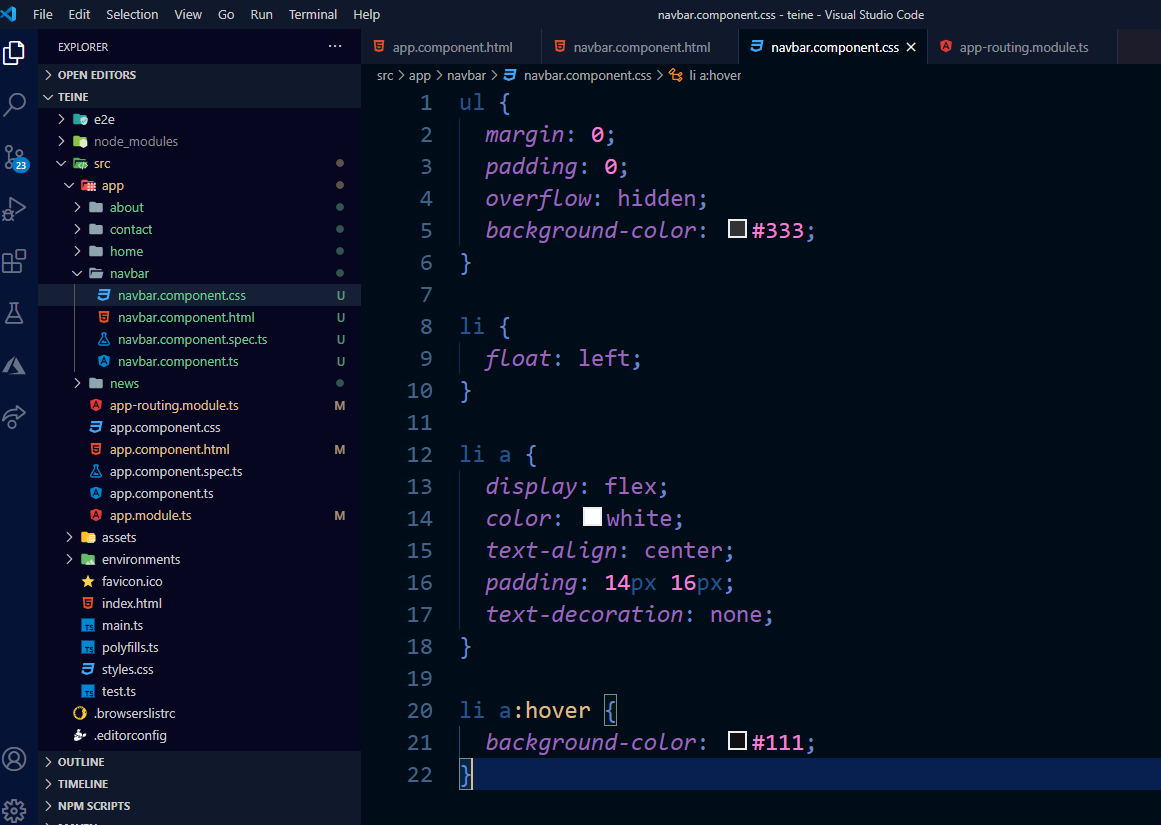


See kood tuleb sinna, kus me võtsime navbari kasutusele ehk app.component.htmli. Seega see kood mis me siia kirjutame on igal leheküljel nähtav.

1. **Mine localhost:4200 peale ja testi, vajutades linke**



1. **Lisa järgnev CSS navbar CSS sisse:**



ul <- unordered list

li <- list item

Vaata ka HTMList.

a <- anchor (HTMLis URL vahetus)

:hover <- hiirega üleminek

Margin <- ümbritsev kaugus väljaspool seda elementi

Padding <- ümbritse kaugus sees pool seda elementi

Võtke margin ja padding ära ja vaadake mis juhtub.

Lisage margin 100px, vaadake mis juhtub.

Pange algolukord ja lisage padding 100px, vaadake mis juhtub.

Vasakult paremale saime elemendid liikuma flex abil.

Text-decoration abil võtsime a elemendiga kaasneva alumise joone ära.

Kui lehekülgi vahetate, jälgige kui sujuvalt see toimub. Refreshi ei juhtu, kuna meil on üks HTML ja ei toimu teisele „leheküljele“ minemist nagu tavaliselt.

Lisage „kodu“ sisse pealehe kujundus.

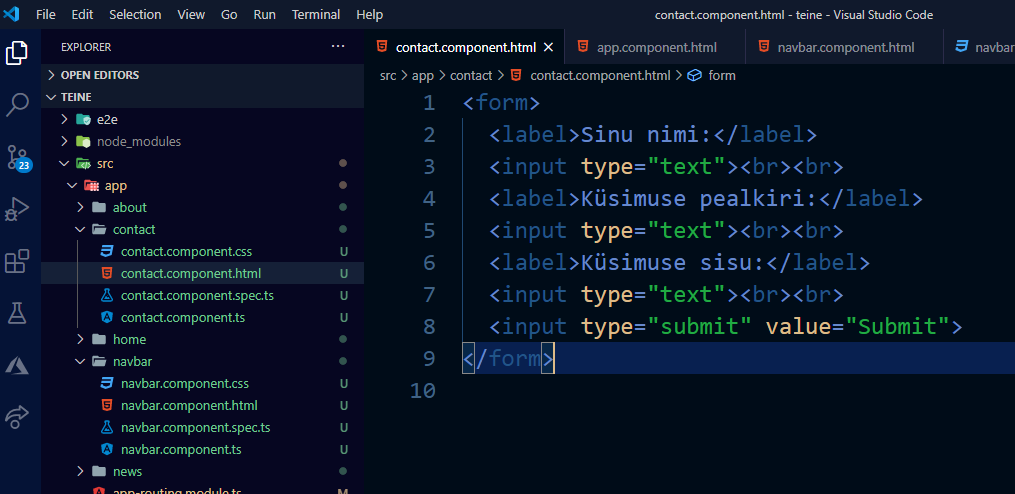
Lisage uudised sisse uudiste kujundus (näiteks piltidena nagu eelmine kord). Võite ka

<ul> <li></li> <li></li> </ul> abil midagi lisada.

Lisage osadele uudistele ka youtube videod.

Meist juurde lisage pilt ja mingi tekst.

Kontakti juurde lisage lihtsalt HTMLi mõtteis:



Tulevikus tegeleme ka vormidega, aga praegu on sellel vaid väljanäitamise väärtus. Kui tahate, võite teha kujundust (class=““ kaudu) või otsida internetist sobivat CSSi kujundust vormide jaoks ja seda siia implementeerida.

Sisestus aga ei toimi ja ei tee midagi ning see on okei – oleme valmis teinud vaid htmli ehk väljanäitamise, mitte funktsionaalsused mis asuvad tulevikus .ts failis.