Tee uus projekt, nimega Portfoolio

Eesmärk avalehel:

Graphical user interface, website

Description automatically generated

1. **Lisa kujunduslik pool**

Lisa kõige üles pilt (*CSS allpool*).

Lisa selle alla väike joon, mis mahutub projektis keskele (*CSS allpool*).

Lisa 3 pilti üksteise kõrvale, kasutades display: flex ja justify-content: space-evenly (*CSS allpool*).

Kõik 3 pilti tuleb alla laadida ja **panna „public“ kausta**. Neid tuleb kasutusele võtta:

*(jutumärgid on siin valed)*

<img src=“/**faili-nimi.png**“ alt=““/> täpselt **sama nimega** nagu public kaustas.

1. **Lisa Routing**

Tee projekti kaustas „*npm install react-router-dom*“.

Lisa **BrowserRouter** <App /> ümber index.js failis.

App.js sisse lisa **<Routes>** võimekus ja erinevad **<Route>** võimalused. Route peab olema Routes sees ja Route näeb välja: <Route path=““ exact element={} />

*Pane tähele, et path järgnevad jutumärgid on valed*

1. **Seosed**

Avalehel (path=“/“) näita punktis nr 1 kirjutatut (3 pilti üksteise kõrval)

Tee veel 3 pathi – nende sisu on sinu valida (seda siin failis pole) – element={<div>Suvaline sisu</div>}

Kokku on seega **4 <Route /> seost** <Routes></Routes> sees – 1 avaleht ja 3 alamlehte.

1. **Avalehel Routide vahetamine**

Tee kõik 3 pilti klikitavaks „<Link to=’’“> abil ja vaata kas URL ja sellega seoses ka tekst vahetuvad.

1. **Tee pages kaust ja sinna sisse 4 lehte**

Avaleht

Ja iga alamlehe võiksid samuti eraldi Page-na teha, näiteks:

Work

Hobbies

Courses

Tõsta varem kirjutatu App.js seest nende lehtede sisse ja võta need Componendid kasutusele App.js sees. Tee import igale lehele. Kontrolli, kas kõik töötab nagu 4. sammu lõpus.

Tee igale lehele tagasimineku nupp - <Link to=“/“><button>Tagasi</button></Link>. Link tuleb alati importida! Jälgi ka, et oleks alati üks ümbritsev element.

*<Link to=“/“><button>Tagasi</button></Link>*

*<div>Töölehe sisu</div>*

Annab errori!! Peab olema sellisel juhul:

*<div>*

*<Link to=“/“><button>Tagasi</button></Link>*

*<div>Töölehe sisu</div>*

*</div>*

Avalehe CSS on järgnev:

<div className=“main-link-list“>

<div className=“main-link“>

<img src=“/public/pilt.jpg“ />

<p>Test</p>

</div>

</div>

Text

Description automatically generated

Tähendused:

Display: flex 🡪 kõik ELEMENDID selle sees on **vasakult paremale**

Justify-content: space-evenly 🡪 need elemendid mis on vasakult paremale, nende vahel olev **ruum (space) jaotatakse** nii, et see on kõikide ümber võrdselt

Katsetage, pannes sinna asemele:

Justify-content: space-between;

Justify-content: space-around;

Color: 🡪 **tekstivärv**, katsetage teiste värvidega

Background-color: 🡪 **taustavärv**, katsetage teiste värvidega

Margin 🡪 **lükatakse** midagi sellest **elemendist** **kaugemale** igas suunas.

Margin-top 🡪 **lükatakse** midagi sellest elemendist kaugemale **ülespoole**. Katsetage pannes 200px;

Padding 🡪 **lükatakse** selle elemendi **sisemust sissepoole**. Kui on 1 väärtus, siis igas suunas, kui on kaks väärtust, siis esimene väärtus on üles-alla ja teine väärtus vasakule-paremale, kui on neli väärtust, siis on üles-paremale-alla-vasakule (nagu kell).

Katsetage Paddingut:

padding: 200px 0;

padding: 20px 20px;

padding: 0 200px;

padding: 200px;

padding: 0 20px 60px 40px;

Border-radius 🡪 mis kohast hakatakse „**kurv**i keerama“ – katsetage 100px;

Kui tahta ringi ristküliku asemel, siis on võimalik kirjutada:

Border-radius: 50%; (keskelt hakatakse „kurvi keerama“)

Text-align: center; tekst läheb **keskele**

Width: **kõrgus**, katsetage suuremate/väiksemate pikslitega

Height: **laius**, katsetage suuremate/väiksemate pikslitega

Hoidke hiirt „**main-link img**“ peal, saamaks aru mis tähendab **tühik** nende vahel

Text

Description automatically generated

Hoidke hiirt „main-link p“ peal, näete sarnast asja. Minge vaadake HTML faili ning näete ka loogikat, miks on niimoodi kirjutatud -> tahame anda stiili paragraphile või image-le.

ME VÕIME ka selle asemel kirjutada className=“main-link-image“ HTMLi ja me ei pea kasutama .main-link img struktruuri, aga me säästame end ühe klassi lisamisest – KUIGI mõlemad viisid töötavad täpselt sama hästi ja korrektselt.

Kuna meil on vaid üks pilt HTMLis, siis võime ka kirjutada lihtsalt img (jätta klassi ära), kuid kui pilte tuleb koodiossa juurde, siis on hea täpsustada, kus täpselt see pilt asub.

Text

Description automatically generated

:hover 🡪 **hiirega üleminek**

transition: .5s 🡪 mitu sekundit võtab allolev koodilõik **aega**. Katsetage pannes:

transition: 1s;

transition: 5s;

transition: .1s;

transform: scale(1.1); 🡪 **transformeeritakse** (muudetakse) olemasolevat 1.1korda suuremaks. Katsetage pannes:

transform: scale(0.6);

transform: scale(6);

cursor: pointer; 🡪 vajutatav käemärk kursoril, Katsetage teisi **kursorimärke**:

cursor: cell;

cursor: grab;

cursor: progress;

cursor: wait; jne

box-shadow: **vari kasti ümber.** Katsetage teiste värvide ja teiste (ekstreemsete) pikslitega

margin: auto; 🡪 auto ehk „**brauser leia ise** margin“, paneb sobivates olukordades keskele.

**Mõned ideed veel:**

1. Youtube video lisamine:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

(kui tahad alustada mingilt kindlalt sekundilt)

Graphical user interface, text, application

Description automatically generated

Kopeeri koodilõik enda HTML sisse. Võid kõike proovida muuta – width, height asemele teised numbrid, allow=““ jutumärkide sees midagi ära kustutada või terve allow ära kustutada.

1. Ikoonid allaossa:

Võta flaticon.com sobivad ikoonid ja lae see alla png failina. Kliki peale sobiliku ikoonile ja vali roheline „png“ ja seejärel „Free download“.

A screenshot of a computer

Description automatically generated with medium confidence

Otsi see üles Downloads kaustast ja lisa see „public“ kausta, näiteks nimega „facebook.png“.

Võta see kasutusele:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated