Projekt „Laptops“

1. Tee uus projekt

Sellega pöördud npm mooduli **npx** poole.

Text

Description automatically generated

1. Pane see käima sisestades järgmised käsud „**cd laptops**“ ja seejärel „npm start“:

„cd“ tähendab lühendit „change directory“ ning see vahetab kausta. Npm start käivitab projekti aadressil localhost:3000 .

Projekti on võimalik sulgeda vajutades „ctrl + c“ terminalis.

Text

Description automatically generated

Et hoida projektis puhtust, meeldib mulle järgnevad failid ära kustutada, mis midagi juurde ei paku:

Graphical user interface, text

Description automatically generated

See viskab errori, kuna tuleb kustutada ka importid

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Võtan kasutusele Routingu, selleks lisan projekti „react-router-dom“ mooduli:

Seda peab kindlasti tegema samas kaustas ehk „laptops“ kaustas.

Text

Description automatically generated

See installeerimine lisab võimekuse liikumaks erinevate URL-del.

Lisan index.js faili <BrowserRouteri>, kindlasti peab ka impordi tegema:

Text

Description automatically generated

Teen kolm erinevat URL aadressi, millele on võimalik minna:

Text

Description automatically generated

Kui lähen manuaalselt localhost:3000/add või localhost:300/all brauserisse kirjutama, siis näen, et on on kirjutatu olemas:

Graphical user interface, text, application, chat or text message

Description automatically generated

Ma aga ei taha manuaalselt seda ise URLi trükkida, URL vahetus peaks toimima nupule vajutades.

Teeme nupud, millega on võimalik URL-l liikuda

Text

Description automatically generated

Kontrolli, kas kõik töötab – kui vajutad ühe ja teise nupu peale, siis URL muutub koos sisuga.

**Avalehe HTMLi** (vt 3 pilti edasi) saaks küll kirjutada App.js real 17 kirje „Avaleht“ asemel, aga kuna tegemist on hoopis teistsuguse loogikaga, mis ei peaks põhilehel olema (kus määrame URL ühendusi), siis tõstame Avalehe loogika siit välja uude komponenti. Selleks teeme „**pages**“ kausta ja sinna sisse uue komponendi „**Avaleht.js**“.

Text

Description automatically generated

Võtame selle App.js sees kasutusele:

Text

Description automatically generated

Tee ka import!

Kõik failid tuleb alati importida.

Kuva avalehel tekst:

Text

Description automatically generated

Lisa kujundus:

Text

Description automatically generated

Tee valmis ka teised kaks komponenti:

A screenshot of a computer

Description automatically generated with medium confidence

Ning nende seosed:

Text

Description automatically generated

Tee valmis vorm, millega on võimalik arvuteid lisada:

Text

Description automatically generated

Label on tekstiväli, mida võetakse kasutusele koos „input“ väljaga.

Input väli on andmete sisestamiseks.

Tee valmis funktsioon, mida on võimalik nupuvajutusega käivitada:

Text

Description automatically generated

Selleks peab ka täpselt sama nimega funktsiooni üles tegema. Ühtegi täheerinevust (kaasa arvatud suur ja väike täht) ei tohi olla.

Tee useState abil sõnumi väljakuvamine. UseState on funktsionaalsus, mis võimaldab muuta mingit kirjet HTML-s pärast nupu vajutamist.

Text

Description automatically generated

Tee ka import!

Pane tähele, et hetkel näidatakse meile sõnumi kohapeal „Lisa auto“.

Text

Description automatically generated

Sinna sõnumi juurde saan ka teisi asju kirjuta, aga seda väljaspool loogelisi sulgusid.

Text

Description automatically generated

Loogelised sulud tähistavad **JavaScripti**. Kui sisestaksin selle kirje „Sõnum“ loogeliste sulgude sisse, siis programm otsiks seda sisu JavaScriptist.

Hetkel programm otsib sõnumit „message“ JavaScriptist ja leiab selle.

Text

Description automatically generated

Kui aga sisestaksime näiteks {Sõnum} loogelise sulgude sisse

Text

Description automatically generated

Programm hakkab otsima JavaScriptist muutujat nimega „Sõnum“, mida ei leita. See väljendub erroris kui avad Terminali:

Text

Description automatically generated

Meile öeldakse täpne veakoht ette: LisaArvuti.js failis, rida 12, Sõnum on mittedefineeritud.

Error on punane.

Kui aga teeme nii nagu just tegime, on kõik korras:

Text

Description automatically generated

Sõnum: on kiri, aga {message} otsitakse JavaScriptist, mis ka leitakse. Täpselt sama kehtib ka funktsiooni puhul:

Text

Description automatically generated

Terminali vaadates on aga Warning (kollane):

Text

Description automatically generated

Rida 4 LisaArvuti.js failis on mingi funktsioon, mida pole kunagi kasutatud:

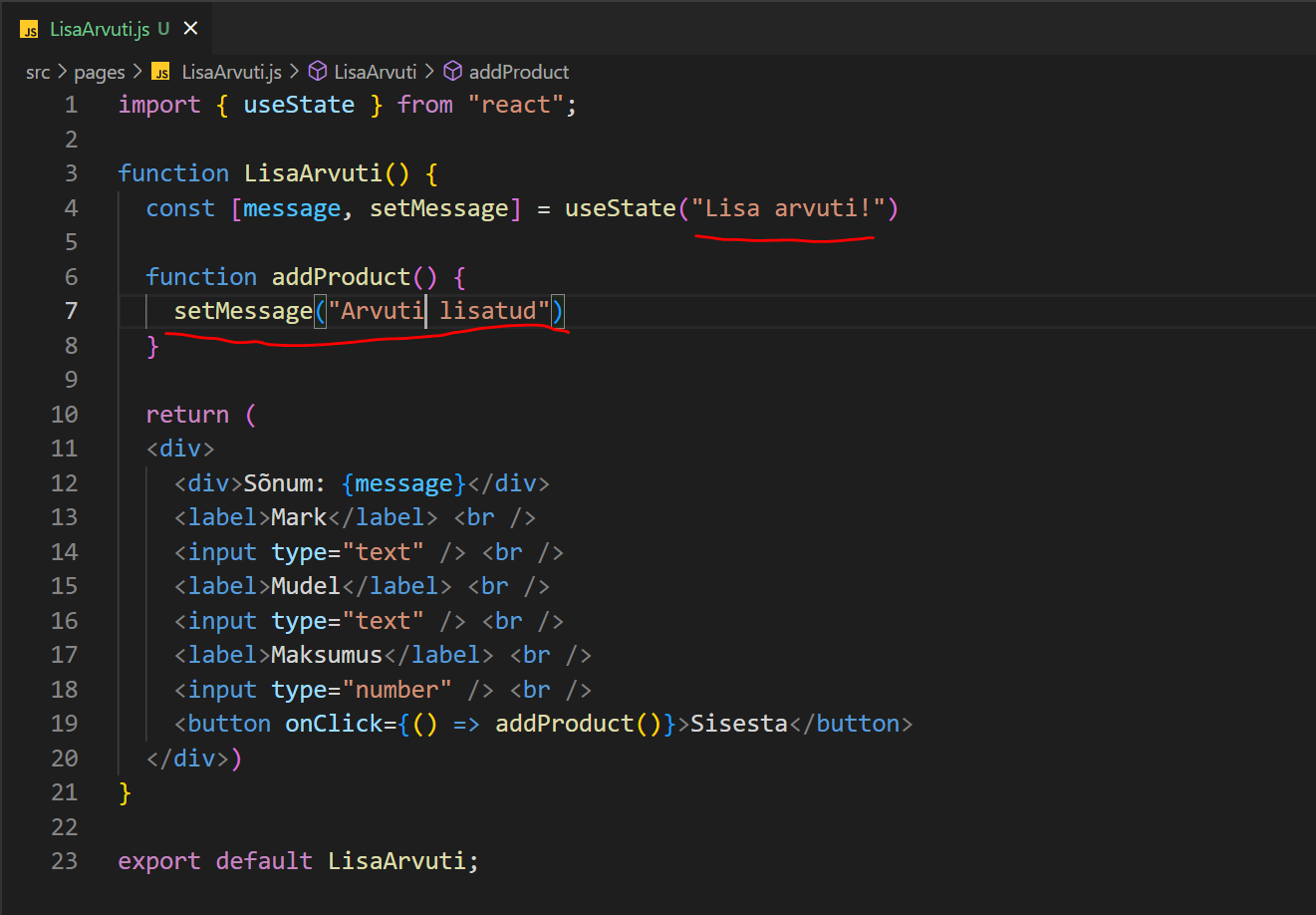
A screenshot of a computer

Description automatically generated with medium confidence

Ta on tõepoolest ka läbipaistvat värvi ning pole kunagi kasutatud.

React ütleb, et kui me midagi ei kasuta, siis sellel pole väärtust ning võiksime selle ära kustutada. Võtame aga selle funktsiooni kasutusele.

Muudame ka algväärtuse ära -> „Lisa auto“ asemele „Lisa arvuti“



Algväärtus on nüüd muutujal „Lisa arvuti“:

Graphical user interface, text, application, email

Description automatically generated

Kui sisestata arvuti, siis muutub „message“ väärtus:

Graphical user interface, text, application

Description automatically generated

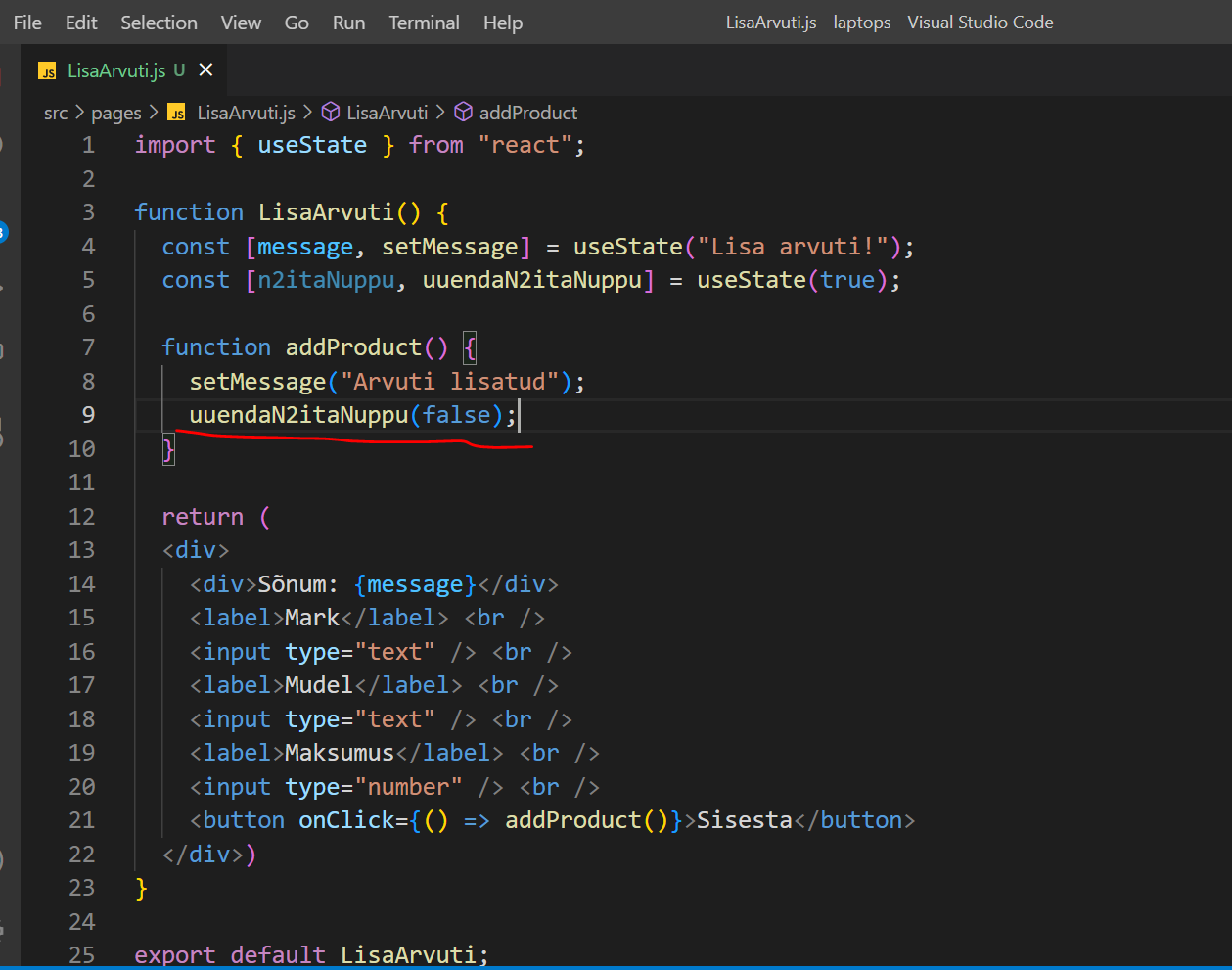
KÕIK, mis lehel muutub, peab muutuma läbi useState funktsiooni. Lehel võib olla väga suur hulk muutuvaid kohti, mis nupuvajutuse või hiireklikiga muutuvad – need tuleb KÕIK Reactis panna useState abil muutuma.

Näiteks kui on soovi panna nupp kaduma pärast sisestamist – teeme uue useState:

Text

Description automatically generated

Siin on sulgude sees *boolean,* mille Eesti keelne vaste on kahendväärtus – see võib olla kas TRUE või FALSE ja mitte kunagi mitte midagi muud. Algväärtus on sulgude sees – see on „TRUE“. Seega nuppu alguses näitame. Kui aga sisestatakse, siis teeme, et ei näidataks:



Nüüd muutus kahendväärtus FALSE ehk väära peale. Teeme dünaamilise väljakuvamise HTMLs. Me peame ütlema tingimuse millal NÄIDATAKSE nuppu. Näidatakse, kui n2itaNuppu muutuja on TRUE:

Text

Description automatically generated

Pane tähele, et vahepeal tekkisid loogeline sulg ette ja taha. See tähistab JavaScripti lisamist:

Text

Description automatically generated

&& tähistab, et kui vasak pool on tõene, siis näidatakse paremat poolt.

Text

Description automatically generated

Nüüd lisades nupp kaob. On dünaamika. Refreshiga või teisele URL-le minnes ja tagasi tulles on kõik aga algseisundis. Meil ei ole salvestust.

Kokkuvõte.

Importida tuleb Reacti kõik erikood. UseState on erikood. useState nimetatakse Reactis HOOK-ks.

Text

Description automatically generated

See on komponendi nimi ja tähistuse algus:

Text

Description automatically generated

Komponent algab loogelisest sulust ja lõppeb loogelise suluga:

Text

Description automatically generated

Export järel pannakse komponendi nimi. Kõik failid on vaja Reactis exportida, sest neid on vaja ka importida.

Text

Description automatically generated

Komponendi (lehe) üleval osas luuakse muutujad, näiteks useState:

Text

Description automatically generated

Neil on kõigil „const“ eesliides ning kandiliste sulgude vahel on:

vasakul pool muutuja, mis sisestatakse HTMLi loogeliste sulgude vahele

paremal pool funktsioon, mida kutsutakse klikkides HTMLs või funktsioonis JavaScriptis välja. See kutsutakse välja alati nii, et tema nime lõppu pannakse sulud ja selle sisse uus väärtus.

useState() sulgude sees on algväärtus – pärast refreshi või URLi vahetades ning tagasi tulles.

Komponendi muutujate all on funktsioonid, mis pannakse HTML-s käima nupuvajutusega:

Text

Description automatically generated

Need funktsioonid käivitatakse HTML-s onClick järgselt, kui sama nimetus välja kutsutakse. Käivitub alati blokk, mis on loogelisest sulust loogelise suluni.

Text

Description automatically generated

Funktsiooni sisesene blokk kutsub välja omakorda ülemised funktsioonid.

Iga rea lõpus olev semikoolon on vabatahtlik, seda ei pea panema, aga tavaliselt seda tehakse, tähistamaks rea lõppu

Text

Description automatically generated

Return osa sees on alati HTML, mida kuvatakse välja sellel lehel

Text

Description automatically generated

See peab kindlasti olema üks komplekt (algama ja lõppema sama elemendiga)

Text

Description automatically generated

DIV on tühjus või mõtteta element – seda lisatakse tekstidele ja ümbritsemiseks:

Text

Description automatically generated

Label ja input on tavaliselt koos – label tabele mis sinna sisestusvälja sisestatakse ning input võtab väärtuseid vastu

Text

Description automatically generated

Text

Description automatically generated

<br /> tähistab uue rea lisamist

Button on nupp

Text

Description automatically generated

Funktsioon millega käivitatakse, käib alati <button **SIIN** > NUPU KIRI </button> ehk nupu avanemise kolmnurkade sees

Text

Description automatically generated