Sisällysluettelo

Local storage

Aloitus 2

Suunnitelma 2  
 -Logiikka 2  
 -Mockups 3

Lomake 4  
-HTML 4  
-CSS 4  
-Javascript 4  
-Local storage 4

Lomakkeen toimivuus 5  
-Lomakkeen testaus 5  
-Local storagen testaus 5  
-Valmis lomake 5

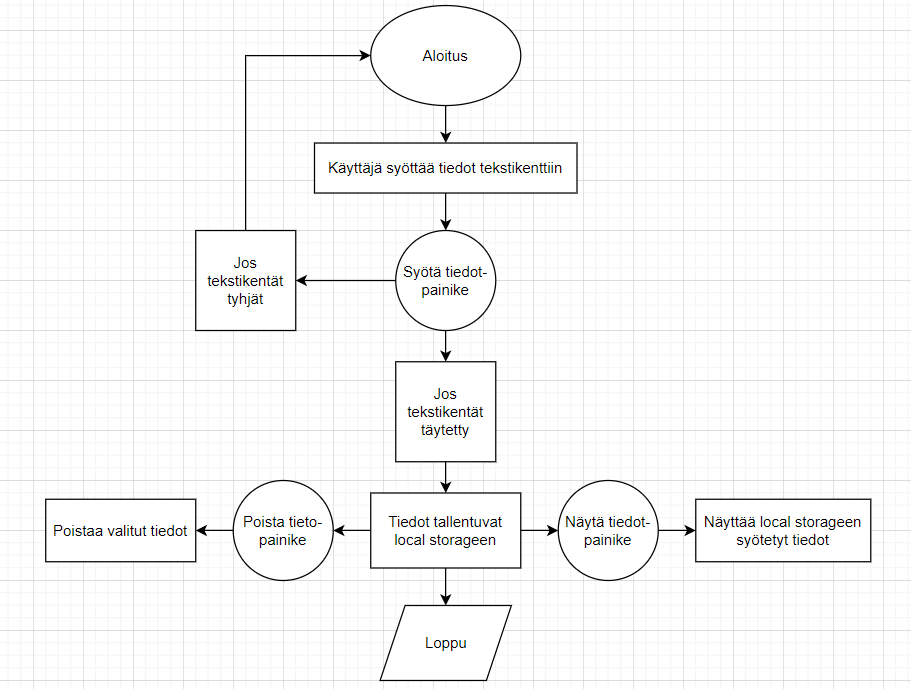
Havaitut virheet 6

Aloitus

Ideana tehdä yhteystietolomake, johon käyttäjä täyttää tietonsa. Täytetyt tiedot tallentuvat local storageen ja kaikki tiedot mitä yhteystietolomakkeella on syötetty ovat saatavilla jälkeenpäin vaikka selaimen sulkee välissä. Painikkeilla pystyy joko tallentamaan, näyttämään tai poistamaan tietoja.

Suunnitelma

Logiikka



Mockups





Web-sivu on jaettu kahteen osaan jolloin erottuu yhteystietolomake sekä haetut tiedot toisistaan.

Lomake

HTML

HTML:ä luodaan web-sivu jossa itse lomakepohja, sen painikkeet sekä tiedot-kenttä sijaitsee. Vasemmalla puolella on tekstikentät johon voi kirjoittaa pyydetyt tiedot. Yhteystietolomakkeen alapuolella on kaksi painiketta. Toinen painike syöttää/tallentaa antamasi tiedot local storageen ja toinen hakee kaikki sinne syötetyt tiedot ja ne näytetään sivun oikella puolella tiedot-kentässä.

Sivun oikealla tiedot-kentässä näkyvät jo aikaisemmin syötetyt tiedot ja poista tiedot-painikkeella sen pystyy poistamaan jolloin se ei enää näy sivulla.

CSS

CSS:ä määritellään web-sivun sekä lomakkeen ulkonäkö. Sillä muokataan taustaväriä, lomakkeen väriä, kirjainkokoa ja fonttia sekä määritelty elementtien sijainnit sivulla.

Javascript

Javascriptillä tehdään funktiot joilla tiedot saadaan tallennettua local storageen, näytettyä ne sieltä sekä poistettua ne sieltä. Funktiot laitetaan nappeihin joilla kyseiset toimenpiteet voi suorittaa. Linkkeinä käytetään HTML sivussa luotuja tekstiruutujen ID:itä.

Local storage

Local storageen syötetyt tiedot tallentuvat selaimen muistiin, jolloin tietoja voi hakea jälkeenpäin vaikka selain on suljettu välissä.

Local storageen tallennetaan kenttiin täytetyt tiedot setItem() metodilla. Sen jälkeen ne haetaan sieltä getItem() metodilla niille luotuun div kenttään. Ja niille luotiin napit joilla ne pystyy poistamaan local storagesta removeItem() metodilla.

Lomakkeen toimivuus

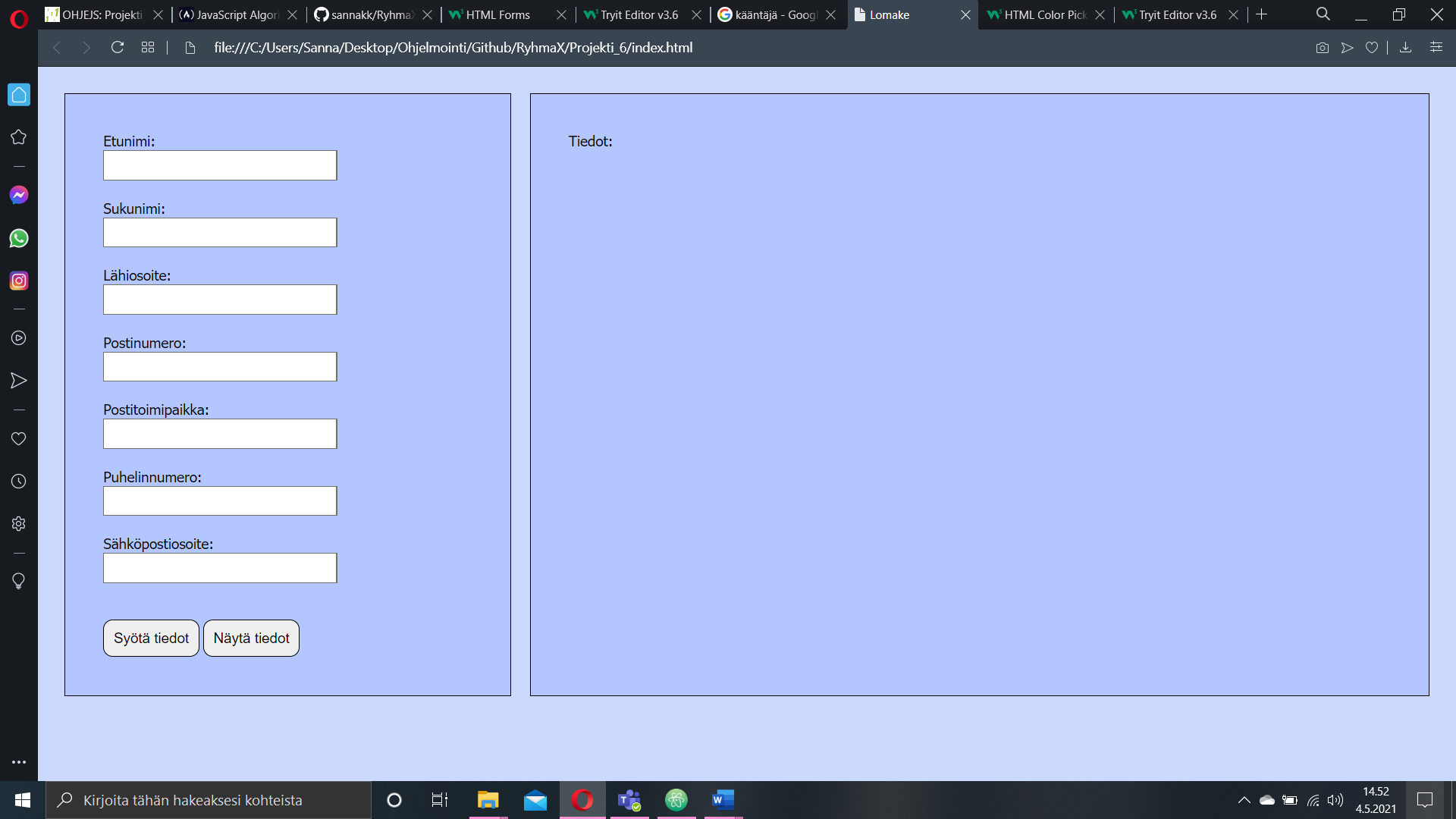
Lomakkeen testaus

Lomaketta testattiin monella eri tavalla. Konsolin kautta testattiin, että mitä tietoja ID:llä haku antaa ja ovatko ne oikein. Tietojen syöttöä ja niiden näkyvyyttä testattiin syöttämällä lomakkeeseen tietoja eri tavoin, jolloin nähtiin miten esim. ”väärin” syötetyt tiedot vaikuttavat lomakkeen toimintaan sekä miten kaikki syötetty teksti näkyvät oikealla tiedot-kentässä.

Local storagen testaus

Loimme konstantteja jotka hakevat kenttiin täytetyt arvot jotta saisimme ne tallennettua local storageen. Sen tehtyämme testailimme miten saisimme ne tiedot tallennettua sinne. Keksimme käyttää if lausetta, eli jos kenttä on täytetty se tallennetaan sen kentän Id:llä local storageen. Aluksi testasimme myös eri tapoja miten saisimme tiedot niin, että ne eivät korvaantuisi. Kun meille selvisi ettei niiden tule korvaantua lopetimme sen testauksen.

Valmis lomake



Havaitut virheet

|  |  |  |
| --- | --- | --- |
| Virhe | Mistä johtui | Virheen korjaaminen |
| Id:llä haetut tiedot antoivat arvoksi ”null”. | Muuttujiin tallennetuista objekteista puuttui perästä .value. | Lisäämällä .value perään, muuttujiin saatiin haettua oikeat tiedot. |
| Viimeisimmäksi lisätyt tiedot korvasivat aikaisemmin syötetyt, vaikka tarkoitus oli että kaikki säilyvät. | local.Storagessa se on tarkoitus, että se korvaa tiedot. Emme ymmärtäneet aluksi, että sen kuuluisi tehdä juuri niin. | Olimme saaneet sen jo aikaiseksi, että tiedot korvaantuu joten jätimme sen niin kun meille selvisi että näin pitääkin tapahtua. |
| Poista tiedot-painike ei poistanut tietoja local storagesta. | Se ei saanut kiinni Id:stä joka piti poistaa. | Teimme jokaiselle napille oman funktion joka poistaa juuri kyseessä olevan Id:n local.Storagesta. |
| Lomakkeeseen pystyi kirjoittamaan kirjaimia numerokenttään. | Inputin tyypiksi oli määritelty ”text”. | Input tyypin vaihtaminen ”number”. |