Alustasin kodutööga 20.12, kui esmalt viisin end põgusalt kurssi uute tundmatute tehnoloogiatega nagu Angular ja Liquibase. Järgmisena lõin tühja Spring Booti projekti ning kasutasin Vertabelo rakendust, et luua SQL skripte tabelite jaoks. Backendiga alustades seadistasin esmalt Liquibase, mille kohta sain infot Youtubest, ning seejärel jätkasin esialgse back end projekti koodiga.

Kuna ma ei olnud mõnda aega täiesti nullist projekte teinud ehk harjumust ei olnud veel, siis laadisin Githubi ülesse tavalisest suuremahulisema backend projekti esialgse versiooni. Veidi rohkem aega kui võiks võttis H2 andmebaasi seadistamine arvutisse ja selle iga tabeli korrektne kuvamine Intellij Database aknas.

Angularis frontendi seadistamise ja komponentide/moodulite/teenuste/navigeerimise tõed sain samuti Youtubest (tundma õppimise ajakulu peamiselt üks õhtu)

Backendi ülesehituses kasutasin sama struktuuri nagu enda Vali-IT koolituse lõpuprojektist, kust võtsin ka errori käsitlemise/valideerimise osa, mapperi seadistamise.

Frontendi ülesehitusel kasutasin samuti enda Vali-IT koolituse lõpuprojekti, kus samuti kasutasin bootstrapi. Siin projektis rakendasin kuvamise poole pealt tabelite ja vormide kujundusi.

Vormi andmete inline valideerimine oli üks keerukamaid ülesandeid, kust abi sain nii stackoverflowst (patterni idee) kui Youtube keskkonnast (Angulari vormi valideerimine). Samas dünaamilist andmete kuvamist ja lisamist sain abil taaskord oma Vali-IT projektist, kus küll frontendi oli Vue raamistikuga tehtud, aga üldpõhimõtted on samad. Eelkõige oli siin pusimist paar tundi konkreetselt “edit pet” funktsionaalsusega, et kuvada select dropdowni juba eelnevalt kasutaja poolt initsialiseeritud väärtus, mida omakorda saaks muuta.

Üldise kujundus tegin pealtnäha lihtsa, et oleks mugav kasutada ja oleks kasutajale selge, kuid kindlasti saaks siin veel lõppviimistleda.

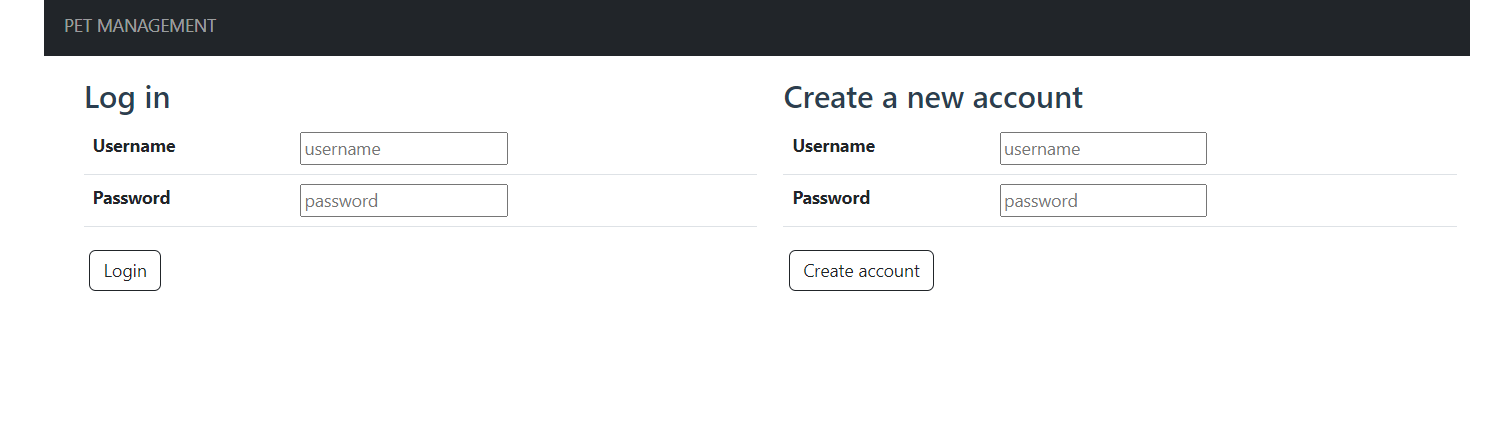
Lisaks tavanõuetele tegin ära kõik optionalis toodud ülesanded: lemmiklooma muutmine, sisse- ja väljalogimise kasutaja järgi, ainult omaniku lemmikloomade vaate kuvamine, sorteerimine tabeli parameetrite järgi, päritolumaa tabeli ja veeru lisamine, vormide valideerimine.

Huvist, vabast ajast ja õppimise soovist tegin ekstrana juurde: uue kasutaja registreerimine koos valideerimisega, admin vaade, kus: 1) näeb ja saab järjestada kõiki loomi 2)näeb kõikide kasutajate andmeid ja lemmikloomade arvu 3)uute selekteerimise väärtuse loomise koos valideerimisega, lemmiklooma kustutamine, automaatse unikaalse lemmiklooma koodi genereerimine ja selle kuvamine.

Autoriseerimise poole pealt pole mu rakendus täpne, sest näiteks olles tavakasutajana sisselogitud jai URLi muutes admini lehel peale, siis see ka sinna viib, aga ei tohiks olla lubatud tegevus. Samas see autoriseerimist ei olnud kodutöös ka nõutud, aga kindlasti peaks sellega tegelema, kui tahta korrektset lahendust.

Kasutasin programeerimiseks kahte arvutit erinevatel aegadel: oma isiklikku Windows op süsteemiga arvutit ja praktikakoha arvutit, milles on Linux op süsteem.

Pet management rakenduse tööle panekuks tuleb esmalt sõltuvused alla laadida kummagi projekti jaoks, seejärel backend clickiga käima panna (roheline play nupp) ja seejärel frontend clickiga käima (või terminalis kirjutada „ng serve –open“). Seejärel peaks avanema järgnev aken:



Admin vaatesse pääseb: „admin“ ja pw:123