

CGI suvepraktikale kandideerimise ülesanne

Ülesandeks on luua lennureisijale lennu planeerimise ja lennukis istekohtade soovitamise veebirakendus.

Reisija soovib lennata sihtkohta ja peab saama rakendada erinevaid filtreid, mille alusel talle lende soovitatakse.

Lisaks peab rakendus soovitama lennukis istekohti, võttes arvesse tema eelistusi (kas ta eelistab istuda akna all, rohkem jalaruumi, lähemal väljapääsule jne). Istekoha sovitamine peab toimuma lennuki istekohtade plaanil. Juba kinni olevad istekohad võiks genereerida juhuslikult, näited, millega kohtade soovitamisel võiks arvestada, on välja toodud allpool.

Tehnoloogiad

Ülesande lahendamiseks ei ole ette nähtud kindlat front-end raamistikku. Rakenduse *back-endis* tuleks kasutada *Spring Boot*'i ning viimast Java LTS versiooni. Tuleb kasutada versioonikontrolli (Git).

Lahenduse etapid

Lennu valimine

Kasutaja peaks alguses nägema lennuplaani kõigi lendudega. Kasutaja peab saama lennuplaani filtreerida, näited filtritest, mida võib rakendada (nimekiri ei ole lõplik)

- Sihtkoht
- Kuupäev
- Lennuaeg
- Hind

Istekohtade sovitamine

Kui lend on valitud, peab kasutajale soovitama istekoha(d), sõltuvalt sellest, mitmele inimesele ta soovib piletid osta. Soovitused tuleks kuvada lennuki istekohtade plaanil. Lennuki suuruse ja istmete konfiguratsiooni võib vabalt valida, juba hõivatud istekohad peaksid olema genereeritud juhuslikult. Siinkohal võiks arvestada järgmiste võimalike filtritega (nimekiri ei ole lõplik)

- Istekoht akna all
- Rohkem jalaruumi
- Lähedal väljapääsule
- Istekohad kõrvuti (kui ostetakse korraga mitu piletit)

„Kui aega jääb üle“

Kui tunned, et tahaks teha midagi veel, et sinu töö eriliselt silma paistaks, on siin mõned ideed, kuid piiriks on sinu fantaasia

- Lisa erinevad istekohtade klassid (1. klass, äriklasse, turistiklass)
- Lae andmed lendude kohta mõne reaalse lennufirma API kaudu.

- Toeta lennu planeerimisel ümberistumisi, kui sihtkohta otselendu ei toimu
- Lisa rakendusele testid
- Paiguta oma rakendus Dockeri konteinerisse, mis väldib vajadust selle käivitamiseks midagi installida

Dokumentatsioon

Meie poolt hindab ja käivitab lahendust inimene, kes pole varem seda arendusprojekti näinud. Tal peaks olema lihtne rakendust käivitada, samuti võiks dokumenteerida, mida ja kuidas tegid.

Pane kirja tööks kulunud aeg ja tee märkmeid selle kohta, mis oli keeruline. Kui jäid mõne probleemi lahendamise puhul jänni, pane kirja, kust said abi ja kuidas probleemi lahendasid. Kui mõni probleem jääb koodis lahendamata, siis kirjelda, kuidas sinu arvates seda võiks lahendada. Kui ülesande püstituses on sinu jaoks mitmeti mõistetavusi, siis märgi ära eeldused, mida oled ülesande lahendamisel teinud.

Soovituslik on koodi *committida* pigem varem ja tihemini, mis annab samuti aimu, kuidas aja jooksul lahenduse valmimiseni jõudsid.

Juhul, kui kasutad lahenduses näidisprojektidest, internetist, *Stackoverflow*'st või AI tööriistadega genereeritud pikemaid koodijuppe, palun viita neile dokumentatsioonis ja kommentaaridega koodis, et oleks võimalik aru saada, milline oli sinu panus.

Proovitöös hindame ülesande sisulist ja tehnilist teostust ning lahenduse dokumenteerimist. Saada lahendus tagasi ka siis, kui ei jõudnud kõike ära teha või see pole sinu arvates veel laitmatu. Kui otsustad lahendust mitte saata, anna ka sellest meile teada, et teaksime sellega arvestada.

Lahenduse esitamine

Ülesande esitamiseks tuleb kasutada avalikku koodihoidlat – näiteks Github, Bitbucket, tagasi tuleb saata lahendusele viitav link.