

Web-palvelinohjelmointi : Harjoitustyön ohje

Esimerkkejä harjoitustöistä:

- <https://student.labranet.jamk.fi/~K8517/Ruokalistat/RAPORTOINTI/raportti.html>
- <https://github.com/Jamppa88/NodeTest/wiki>
- <https://student.labranet.jamk.fi/~K8623/doc/>

Kuvaus

1. Harjoitustyössä rakennetaan opintojakson aihealueeseen liittyvä palvelinpuolen web-sovellus **PHP**-ohjelmointikielellä (myös **Python/Perl/Ruby/NodeJS** tai muu vastaava käyvät niin sovittaessa).
2. Harjoitustyöprosessissa opiskelija oppii, soveltaa ja osoittaa opintojakson aihealueen hallintaa laajemmin ja syvemmin kuin pelkästään harjoitustehtävien avulla. Parhaimmillaan tämä näkyy sellaisten ratkaisujen soveltamisessa, jotka eivät ole suoraan nähtävissä opintojaksolla tarjotuista materiaaleista.
3. Harjoitustyön suunnittelu- ja tekoprosessi sekä tuotos dokumentoidaan HTML-muotoisella harjoitustyödokumentaatiolla.
4. Harjoitustyö tehdään optimissaan **1-3** hengen ryhmässä. Poikkeava ryhmäkoko on hyväksyttävä ennen työn aloitusta. Jokaisen työntekijän tekemä työ on dokumentoitava selvästi.
5. Harjoitustyön aihe voidaan valita melko vapaasti, mutta aihe kannattaa **tarkistuttaa** kurssin opettajalla ennen tekemään ryhtymistä. Aihe on helpointa tarkistuttaa laatimalla alustava suunnitelma (**) ja esittämällä se opettajalle.
6. **Työaika** on varattu harjoitustyöhön n. **40** tuntia per henkilö, mutta tämä määrä voi ylittyäkin.
7. Työ esitellään **harjoitustyöseminaarissa** erikseen sovittavan aikataulun mukaan. Esityksen voi pitää myös videoesittelynä. Verkko- ja monimuotototeutuksissa esittely suoritetaan aina videoesittelynä. Videoesittelyn pituus on 3-10min. Esittely on pakollinen.
8. Harjoitustyötä työstetään omalla ajalla. Lähiopetusjaksoja sisältävissä toteutuksissa ohjausta voi saada myös lähiopetuksen yhteydessä.

9. Työn **lähdekoodi** ei voi olla salaista, vaan se pitää toimittaa työn loputtua dokumentaation mukana.
10. Pyydettyäessä opiskelijan tulee pystyä kertomaan harjoitustyössä käytettyjen ratkaisujen keskeisestä teknisestä toimintalogiikasta esim. (video)puhelun välityksellä. Harjoitustyö kaikkineen tulee olla saatavilla siihen asti, kunnes opintojakson suoritus on virallisesti hyväksytty ja kirjattu opintotietojärjestelmään.

Palautus

1. Palautus: **Ks. palautuspäivämäärä kurssitoteutuksesta.**
2. Työn valmistumisesta ilmoitetaan sähköpostilla kertomalla ainoastaan URL, josta työ löytyy. Työn tulee olla tuolloin valmis ja opettajan tarkastettavissa. Mitään ZIP-paketteja ei postiteta.
3. Palautuksen muotoseikoista. Yhden **URLin** takaa löytyy:
 - Työn loppudokumentaatio: Harjoitustyön nimi, tekijöiden nimet, päivämäärä ja tieto siitä, minkä kurssin harjoitustyö on kyseessä.
 - Tehtävän kuvaus
 - Käytännön toteutuksen selostus
 - Sovelluksen kokonaisrakenne mielellään graafisesti, luokkarakenne, funktioiden suhteet, sovelluksen suhteet tiedostoihin, tietokantoihin ...
 - Funktiot ja luokat: ainakin tärkeimpien toiminta dokumentoitava, välitettävät parametrit
 - Tietovarastot: tietokannat, tiedostot.
 - Lähdekoodit - Sovelluksen lähdekoodit on oltava selattavissa Web-selaimella.
 - Ajan käyttö - Selvitys työhön käytetystä ajasta ja tekijöiden roolista harjoitustyön tekemisessä sekä työn jakaantumisesta ryhmän sisällä.
 - Itsearvio – oma arvio työstä muutamalla virkkeellä. Arvosanaehdotus/tekijä. Onnistumiset, puutteet, huomioita. Opintojakson arvioiva opettaja ottaa lopullisessa arvioinnissa myös huomioon sen kuinka realistinen oma arvosanaehdotus on.
4. Linkki kaiken edellisen sisältävään **zip**-pakettiin.

Harjoitustyön tulee siis olla kokonaisuudessaan verkossa tarkastettavissa yhden URL-osoitteen takana. Suosituksena on, että työ tehdään LAMP-pohjaisella *student.labranet.jamk.fi* -palvelimella tai ainakin lopullinen arvosteltavaksi jätettävä versio julkaistaan siellä. Mahdollista on myös käyttää jotakin omaa palvelinta.

Valmiita esimerkkiaihteita

Vaihtoehto 1: Kurssitarjotinsovellus

Tietotekniikan koulutusta tarjoavan yrityksen "kurssitarjotinsovellus", joka jakaantuu a) julkiseen kaikille näkyvään osaan ja b) sisäisesti käytettävään hallintatyökaluun, jolla kursseja ja kurssiryhmiä (esim. Windows 7, Linux, Qt, Android...) voidaan lisätä, poistaa ja editoida. Asiakkaille tulee olla ainakin alkeelliset mahdollisuudet hakea kursseja sopivin hakusanoin.

Sovelluksen "erikoisominaisuus" on se, että julkinen kaikille näkyvä osa generoidaan staattiseksi cronin (tai muun ajastusohjelman kanssa) avulla esim. kerran vuorokaudessa tai tarvittaessa. Julkista osaa käyttäville asiakkaille tarjotaan siis aina esiprosessoituja dokumentteja, jotka saadaan ladattua palvelimelta nopeammin selaimelle.

Vaihtoehto 2: Kalenterisovellus

Useamman käyttäjän kalenterisovellus, jonka avulla mille tahansa päivälle voidaan tehdä merkintöjä. Merkintöjä voidaan muokata ja poistaa myöhemmin. Käyttöliittymänä näytetään halutun kuukauden kalenteri, josta merkintöjä sisältävät päivät erottuvat. Näytettävää kuukautta voidaan vaihtaa esim. valintalistan avulla. Esimerkiksi kalenterin alapuolella on tekstialue valitun päivän merkintöjen lisäämiseksi, muokkaamiseksi tai poistamiseksi.

Vaihtoehto 3: Tilanvarausjärjestelmä

tilanvaraussovellus: Varattavia opetusluokkia voidaan lisätä, muokata tai poistaa mille tahansa päivälle tunnin ajanjaksoina: 8-9, 9-10, ..., 19-20. Varaus sisältää varausajankohdan lisäksi varattavan tilan, varauksen tekijän ja käyttötarkoituksen (esim. opetuksen aihe: "WWW-palvelinohjelmointi"). Käyttöliittymänä voidaan valinnan mukaan näyttää halutun kuukauden tai viikon kalenteri, josta merkintöjä sisältävät päivät erottuvat/tiedot näytetään. Näytettävää kuukautta/viikkoa voidaan vaihtaa

esim. valintalistan avulla. Varausten lisäys, muokkaus ja poisto voidaan toteuttaa halutulla tavalla. Esimerkki <http://tilanvaraus.jamk.fi>.

Vaihtoehto 4: Kiinteistöjen myynti- ja vuokraus

Tehdään Web-palvelu, jossa hallintatyökalun avulla voidaan lisätä/muokata/poistaa myytäviä ja vuokrattavia asuntoja. Ilmoituksien tueksi on voitava ladata kuva/kuvia kohteista, jotka sovellus skaalaa ennalta määritellyn kuvan standardikokoon (pienennetty versio kuvasta).

Sovellus jakaantuu **a)** julkiseen kaikille näkyvään osaan ja **b)** sisäisesti käytettävään hallintatyökaluun. Julkisessa osassa on voitava tehdä kohteista erilaisia hakuja esim. asuntotyyppin, koon tai hinnan perusteella.

Vaihtoehto 6: Paikallislehden verkkoliite

Paikallislehden dynaaminen "verkkoliite". Verkkoliite sisältää sekä uutisia että kolumneja, joita voidaan lisätä/muokata/poistaa.

Sovellus jakaantuu a) julkiseen kaikille näkyvään osaan ja b) sisäisesti käytettävään hallintatyökaluun. Julkisessakin osassa uutis- ja kolumniarkistoon voidaan tehdä hakuja.

Vaihtoehto 6: Verkkotietokilpailu

Mahdollisia ominaisuuksia:

- Kysymykset vaihtoehtokysymyksiä
- Vastausvaihtoehtoja voi olla kahdesta viiteen 2-5
- Kysymyksiä voi olla eri aihealueista (esim. urheilu, historia, tietotekniikka)
- Voidaan muodostaa kyselyitä ylläpitäjän toimesta
- Kysymykset voidaan myös arpoa
- Pisteiden laskeminen
- Vastaaajan tietojen tallentaminen (myös pisteet)
- peli voi olla myös esim. Trivial pursuit -tyyppinen.

ks. esim. <http://www.alypaa.com/>

Vaihtoehto 7: Kauppapaikkasovellus.

Yksi yleisiä harjoitustyön aiheita (tosielämässä kauppapaikkasovelluksia on tehty jo riittävästi, mutta aihe on silti hyvin sopiva kurssin harjoitustyöksi).

Ominaisuuksia ovat mm. selailtavat tuoteluettelot tuoteryhmittelyineen, ostoskori sekä mahdollinen palaute/kommentointilomake. Ylläpitäjälle tarjotaan tilausten hallinta. Esim. <http://www.verkkokauppa.com/>

Vaihtoehto 8: Jalkapallon sarjataulukko ja ottelutilastot

Mahdollisia ominaisuuksia:

- Tallennetaan tilastotietoa jalkapallo-otteluista.
- Ottelusta tallennetaan perustietoina ainakin joukkueet ja tehdyt maalit (myös puoliaikatulos), pelipaikka, tuomaritiedot jne.
- Tallennetaan myös ottelun maalintekijät, syöttäjät, varoitettut pelaajat ja joukkueiden kokoonpano sekä mahdollisuuksien mukaan muita tietoa.
- Tallennetuista tiedoista voidaan generoida sarjataulukko sekä parhaat maalintekijät, syöttäjät, punaiset kortit jne..
- Jalkapallon sijasta voidaan käyttää tilastointimateriaalina myös muita pelejä (esim. Jääkiekko, Koripallo, Lentopallo)
- On myös mahdollista, että sovellus hakee tarvittavan datan esim. Veikkausliigan sivuilta automaattisesti (toteuttaminen vaatii kuitenkin paljon tekstidatan parsimista RE-lausekkeilla), mutta tämä on melkoisen haastava toteuttaa.

Vaihtoehto 9: Itse valittava aihe

Muita aiheita on kerätty tähän. Opiskelijat voivat itse tarkentaa tai keksiä lisää aiheita.

- Opiskelijaportaali, josta saa nopeasti tietoa opiskelijaa hyödyttävää tietoa. Esim. Tietoa kursseista, ruokaloista, ravintoloista jne. Voi laajentaa koskemaan myös esim. jotain opiskeluryhmää.
- Henkilökohtainen palvelinsovellus, joka pitää kirjaa sopimuksistasi, vakuutus- ja pankkiasioista, sähkösopimuksista, vesimaksuista, lehtitilauksistasi yms.
- Joku tyypillinen rekisterisovellus: (CD, Video, DVD, Osoitekirja, Auto)
- tiedostojen hallinta ja jako (kuvat, videot, mp3, dokumentit)

- Taloyhtiölle tuleva portaali, johon tulee kokouspöytäkirjat, piirustukset, sopimukset, yhteystiedot, viranomaisten tiedot, pelastussuunnitelmat jne.
- Grafiikkasovellus (esim. tuotetaan dynaamisesti kaavioita ja diagrammeja datasta)
- mikrotuen avuksi palvelinpuolen statistiikka sovellus (esim. palvelinten kuormituksen seuranta tai HTTP-liikenteen lokien parsiminen)
- Lämpötilojen tilastointi -sovellus. Sovellus tallentaa tietyllä ajan hetkellä olevia lämpötilalukemia relaatiotietokantaan hakien arvon esim. Ilmatieteenlaitoksen Jyväskylän paikallissään näyttävältä Web-sivulta säännöllisin väliajoin. Tallennettuihin arvoihin voidaan sitten kohdistaa erilaisia hakuja tyyliin: "Näytä heinäkuun korkein lämpötila" tai piirtää viivadiagrammia annetulta ajanväliltä.
- Harjoitustyö voi olla myös fokukseltaan erilainen; Se voi esimerkiksi esitellä jonkin:
 - PHP-sovelluskehityksen käyttämistä (Zend, ...) rakennettavan esimerkkisovelluksen avulla
 - PHP-IDEn käyttämistä (PhpStorm, ...) rakennettavan esimerkkisovelluksen avulla

Arviointiin vaikuttaa

- 65% Toteutetuksen lopputulos: Käytetyn tekniikkakirjon laajuus ja laatu (laajempi sisältö antaa mahdollisuuksia laajemman kirjon esittelylle), käyttöliittymän käytettävyys
- 25% Harjoitustyödokumentaatio: laajuus ja laatu
- 10% Työn esittely
- Myös prosessia ja sen suunnitelmallisuutta arvioidaan osana edellä mainittuja kohtia. Kokonaisarvioinnissa huomioidaan myös erityisesti se, kuinka hyvin opiskelija kykenee esittelyssä ja mahdollisessa haastattelussa tuomaan osaamistaan esille harjoitustyön avulla. Opiskelijan tulisi erityisesti tuoda esiin: a) mitä on itse tehty, b) mikä on omasta mielestä ansiokkainta ja c) mitä kehitettävää jäi
- Yksittäisiä harjoitustyössä arvoitettavia asioita:
 - Lähdekoodin kommentointi,
 - muuttujien, funktioiden ja luokkien nimeäminen lähdekoodissa,
 - lähdekoodin luettavuus ja jakaminen tiedostoihin
 - syötteiden tarkistaminen
 - tietovarastojen käyttö (tietokannat, tiedostot, ulkopuoliset tietolähteet)
 - erityisominaisuudet (esim. opintojaksolla läpikäymättömän kirjaston hyödyntäminen)
 - tietoturvan huomioiminen tarvittaessa

- käyttöliittymän selkeys
- oma-aloitteisuus ongelmien ratkaisemissa
- dokumentaation selkeys keskeisten ansioiden ja kehityskohteiden esiintuomisessa

**) Suunnittelusta

Harjoitustyöprosessin alkuvaiheessa kannattaa kirjoittaa ytimekäs vapaamuotoinen suunnitelma, jossa voi ottaa kantaa mm. seuraaviin asioihin

- harjoitustyöryhmän kokoonpano
- aihe (kuvitteellinen, itselle tai jollekin PK-yritykselle tai yhteisölle), kenelle työ on suunnattu (kohderyhmä)
- sovelluksen keskeiset toiminnot
- käyttöliittymä
- käytettävät tietovarastot
- erityistoiminnallisuudet, ulkopuolisten kirjastojen käyttö, ...
- ryhmän työnjaon suunnittelu
- aikataulun suunnittelu jne...

Suunnitelman ja mahdollisesti sen tarkentuneiden versioiden liittäminen harjoitustyödokumentaatioon lisää sen arvoa.