

Web-Kehitys semifinaalitehtävä

Tehtävän suorittamiseen tarvittavat tiedostot ovat saatavilla GitHubissa:

<https://github.com/Taitaja2025/semifinal>

Palvelimen, tietokantojen ja tuotantoympäristön URL-osoitteisiin liittyvät tekniset tiedot löytyvät erillisestä dokumentista. Jokaiselle kilpailijalle on määritetty oma URL ja tunnukset palvelimelle. Arviointi suoritetaan annetussa URL-osoitteessa, ja koodin laatu tarkistetaan kilpailijan Git-repositorystä.

Tehtävän yleiskuvaus

Semifinaalin tavoitteena on toteuttaa toimiva ja käyttäjäystävällinen kysymyspeli, jossa opettajat voivat hallinnoida kysymyksiä ja opiskelijat testata osaamistaan eri aihealueilla. Sovelluksen tulee sisältää kaksi päätoiminnallisuutta: pelinäkymä opiskelijoille ja hallintapaneeli opettajille. Peli on suunnattu kaikille ikäryhmille ja koulutustasoille, joten sen tulee olla monipuolinen ja palvella erilaisia käyttäjäryhmiä.

Tämä kysymyspeli on ensimmäinen versio tuotannosta, joten ratkaisun tulee olla selkeä, hyvin organisoitu ja avoin tulevaisuuden kehitystä varten. Käyttäjäkokemuksen tulee olla sujuva ja visuaalisesti miellyttävä sekä toimia niin opettajien kuin opiskelijoiden tarpeisiin. Kaikki työ arvioidaan toimivan verkkosivuston perusteella, joka julkaistaan sinulle annettuun yksilölliseen URL-osoitteeseen. Tarkemmat tekniset ohjeet löydät kilpailun teknisestä dokumentista.

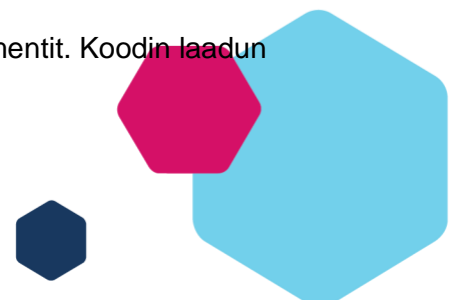
Projektin tekniset vaatimukset

Sovelluksen toteutuksessa kilpailija voi valita käyttää joko natiiviratkaisuja (vanilla HTML, CSS, JavaScript ja PHP) tai moderneja JavaScript-frameworkeja, kuten React, Vue.js tai Angular. Backend-toteutus on tehtävä PHP:llä, ja siinä voi käyttää natiiviratkaisuja tai Laravel-frameworkia. Kaikkien ratkaisujen tärkein tavoite on, että sovellus toimii virheettömästi annetussa URL-osoitteessa. Toteutuksen tulee myös noudattaa hyviä ohjelmointikäytäntöjä ja täyttää tehtävänannon vaatimukset. Graafinen ohjeistus ja rautalankamalli löytyvät **assets**-kansioista.

Käyttöliittymän tyylit ja komponentit on toteutettava ilman valmiita CSS-kehyksiä tai UI-kirjastoja (esim. Bootstrap, Tailwind, Material UI). Kaikki tyylit ja käyttöliittymän komponentit tulee suunnitella ja toteuttaa itse.

Tekoälyn käyttö on sallittua, mutta tekoälytyökaluja hyödyntäessä kilpailijan tulee ilmoittaa nämä kohdat koodissa selkeästi kommentoimalla. Lopullisen ratkaisun on kuitenkin oltava kilpailijan itsenäisesti toteuttama ja omaperäinen.

Koodin tulee olla selkeää, hyvin järjestettyä ja sisältää tarvittavat kommentit. Koodin laadun tarkistamiseen käytetään yhdistelmää työkaluista ja menetelmistä.



Etusivu

URL: /

Etusivun tarkoituksena on olla käyttäjän aloituspiste, jossa tärkeimmät toiminnot ovat helposti löydettävissä. Etusivulla on näkyvä "Pelaa nyt" -painike, joka ohjaa käyttäjän suoraan pelin aloitukseen. Aloituksessa opiskelija valitsee opettajan, kenen kysymyksiin hän haluaa vastata. Lisäksi etusivu sisältää lyhyen kuvauksen pelin toiminnasta sekä opettajien kirjautumislinkin. Sivun alaosassa, footerissa, näytetään tekijätiedot ja oppilaitoksen nimi.

Etusivun ulkoasun tulee olla moderni, selkeä ja saavutettava kaikilla laitteilla. Graafinen ohjeistus ja rautalankamalli löytyvät **assets**-kansioista.

Pakolliset elementit etusivulla:

- **Header:** Pelin logo ja linkki "Kirjaudu sisään".
- **Pelaa nyt -osio:** Lyhyt kuvaus ("Valitse opettaja ja aihealue, ja aloita tietosi testaaminen heti!") ja painike, joka ohjaa pelin aloitusnäkömään.
- **Miten pelataan -osio:** Selkeät ohjeet pelin toiminnasta (kategoriat, opettaja, kysymysten määrä, tulokset).
- **Footer:** Tekijätiedot (Etu- ja sukunimi | Oppilaitoksen nimi) sekä maininta Taitaja 2025 -semifinaalitehtävästä.

Pelinäkymä

URL: /game

Peli käynnistyy, kun käyttäjä siirtyy etusivulta pelinäkymään painamalla "Pelaa nyt" -painiketta. Käyttäjän tulee tehdä kaksi valintaa: ensin opettaja ja sitten aihealue (kategoria).

Kysymysten määrä voi olla lyhyt (5 kysymystä), keskipitkä (10 kysymystä) tai pitkä (15 kysymystä). Oletusvalinta on 10 kysymystä. Jokaisessa kysymyksessä on neljä vastausvaihtoehtoa, joista yksi on oikea. Oikea vastaus antaa yhden pisteen.

Pelin aikana käyttäjä saa reaaliaikaisesti palautetta jokaisesta vastauksesta. Oikeasta vastauksesta näytetään kannustava animaatio, ja väärän vastauksen kohdalla näytetään oikea vastaus visuaalisesti korostettuna. Sinun tehtäväsi on suunnitella animaatiot, jotka tukevat pelaajaa ja sopivat kohderyhmälle. Pelin tulee näyttää pelaajan pistemäärä ja edistymistiedot (esim. 2/10 kysymyksestä vastattu).

Pelin lopussa käyttäjältä kysytään nimi, ja tulokset tallennetaan tietokantaan. Tulokset esitetään high score -listassa, joka järjestetään pisteiden ja vaikeustason mukaan. Lopuksi käyttäjälle näytetään "Pelaa uudestaan" -painike.

Etusivun ja pelinäkymän tulee olla mobiiliresponsiivinen, sillä se on suunnattu laajalle käyttäjäkunnalle, joka saattaa käyttää eri päätelaitteita.

Opettajan hallintapaneeli

URL: /admin

Hallintapaneelissa käytetään valmiiksi annettua tietokantaa, jossa on oletuksena kaksi opettajaa ja valmiiksi luotuja kategorioita sekä kysymyksiä. Kilpailijan tulee käyttää tätä tietokantaa ja tarvittaessa lisätä siihen tarvittavat taulut tehtävän toteutusta varten. Kaikki sovelluksessa käsiteltävä data haetaan tietokannasta ja tallennetaan sinne. Hallintapaneeli on optimoitu työpöytänäköymää varten eikä sen tarvitse toimia mobiilissa.

Hallintapaneeli on suojattu salasanalla, joten käyttäjän tulee kirjautua sisään, ennen kuin hän pääsee hallinnoimaan kysymyksiä ja kategorioita. Kirjautumislogiikan toteutuksessa on varmistettava, että ilman kirjautumista hallintapaneeliin pääsy on estetty.

Kirjautumisnäköymä

URL: /login

Kirjautumisnäköymä on yksinkertainen ja tarkoitettu opettajan hallintapaneelin käyttöön. Näköymän tulee sisältää kentät käyttäjätunnukseksi ja salasanalle sekä kirjautumispainike. Oletustunnukset ovat käyttäjälle valmiiksi määritellyt:

käyttäjätunnus: Olevi Opettaja

salasana: Taitaja2025!

Kirjautumisen jälkeen käyttäjä siirtyy hallintapaneeliin. Mikäli kirjautuminen epäonnistuu, käyttäjälle tulee näyttää selkeä virheilmoitus.

Hallintapaneelissa opettaja voi:

- Luoda uusia kategorioita tai valita olemassa olevia.
- Tarkastella valitun kategorian kysymyksiä ja lisätä uusia kysymyksiä.
- Muokata olemassa olevia kysymyksiä sekä poistaa niitä
- Kirjoittaa uusia kysymyksiä ja vastausvaihtoehtoja sekä merkitä oikean vastauksen.

Jos opettaja valitsee valmiin kategorian, ohjelma näyttää siinä jo olevat kysymykset ja vastaukset ja tarjoaa mahdollisuuden lisätä uusia kysymyksiä. Jos opettaja luo uuden kategorian, siihen voidaan heti lisätä kysymyksiä. Kun opettaja lisää uuden kysymyksen, hänen tulee kirjoittaa kysymys ja merkitä, mikä vastausvaihtoehdoista on oikea.

Kaikkien kategorioiden tulokset ovat nähtävissä hallintapaneelissa. Tulokset voidaan järjestää joko pisteiden tai päivämäärän mukaan. Jos tuloksia on yli 10, näköymä tarjoaa sivutuksen (pagination).

Kaikkien kilpailijoiden on käytettävä annettua tietokantaa ja säilytettävä sen valmiit tiedot. Valmiit kategoriat ja opettajan tiedot tukevat pelin ja hallintapaneelin toimintaa sekä tehtävän arviointia.

Projektinhallinta ja dokumentaatio

Kilpailutyön tärkein tavoite on saada kaikki päätoiminnallisuudet toimimaan annetussa aikataulussa ja julkaistua verkkosivustona sinulle annettuun yksilölliseen URL-osoitteeseen. Ratkaisun tulee sisältää kattava dokumentaatio, joka tukee tehtävän arviointia ja osoittaa toteutuksen selkeyden. Työn tulee olla siistiä, hyvin kommentoitua, ja kaikkien toiminnallisuuksien on toimittava virheettömästi. Kaikki koodin tarkistukset tehdään kilpailijan Git-repositorion kautta. Tarkemmat tekniset yksityiskohdat löytyvät kilpailun teknisestä dokumentista.

GitHub-repositorioon tulee lisätä kattava README.md-tiedosto, jossa kuvaat projektin tarkoituksen, rakenteen ja käyttöohjeet. Selitä, miten peliä pelataan ja kuinka hallintapaneelia käytetään, ja tarjoa ohjeet tuomaristolle testaukseen ja tarkistukseen, mukaan lukien kirjautumistiedot ja tietokantayhteydet.

Koodin tulee olla selkeästi jäsenneltyä ja hyvin kommentoitua. Kommentit tulee lisätä erityisesti monimutkaisiin kohtiin, jotta työn logiikka on helposti ymmärrettävissä. PHP-koodin tulee noudattaa hyvää ohjelmointitapaa, kuten modulaarisuutta, uudelleenkäytettävyyttä ja tietoturvaa. HTML- ja CSS-koodin tulee olla semanttisesti oikein ja validointivirheiden korjattuja. Lisäksi koodin tulee olla optimoitu saavutettavuutta ja suorituskykyä varten.

Tehtävän arviointi

Tehtävän arvioinnissa painotetaan erityisesti sovelluksen toiminnallisuutta, koodin laatua, käyttöliittymän käyttäjäystävällisyyttä sekä dokumentaation kattavuutta. Ratkaisun tulee olla sujuva ja vastata sekä opettajien että pelaajien tarpeisiin, samalla ollen helposti ymmärrettävä ja jatkokehitykseen soveltuva. Kaikkien sovelluksen päätoimintojen on toimittava virheettömästi ja täytettävä tehtävän vaatimukset. PHP-, HTML-, CSS- ja JavaScript-koodin tulee olla selkeää, jäsenneltyä ja validoitavissa ilman virheitä. Käyttöliittymän tulee olla saavutettava, visuaalisesti miellyttävä ja helppokäyttöinen eri käyttäjäryhmille. Dokumentaation osalta README.md-tiedoston tulee tarjota kattava kuvaus projektista, sen rakenteesta ja toiminnallisuuksista, ja sen tulee sisältää selkeät ohjeet tuomaristolle testauksen ja arvioinnin tueksi. Tekoälyn käyttö on sallittua, mutta tekoälyn tuottamat osiot on merkittävä selkeästi koodikommentteihin, ja lopullisen ratkaisun on oltava kilpailijan itsenäisesti luoma ja omaperäinen.