**Frontend – pažengusių EGZAMINAS**

**HTML, CSS, JavaScript, React.js, Node.js, MongoDB**

Egzamino trukmė: **16 val.**

Baigiamojo projekto egzamino sprendimo pateikimas:[**GitHub**](https://github.com/)([instrukcija kaip patalpinti](https://www.youtube.com/watch?v=SWYqp7iY_Tc)) nuoroda\*

*\*siunčiama egzaminuojančiam dėstytojui*

**Galutinio balo sudėtis**

1. **Teorinė dalis ir testas** *(sudaro 10% galutinio balo, trukmė: 30 min.)[[1]](#footnote-0)*
2. **Baigiamasis projektas** *(sudaro 90% galutinio balo, trukmė: 15 val. 30 min.)*[[2]](#footnote-1)

**Teorinė dalis ir testas** *(vertė 10% galutinio balo)*. Trukmė: 30 min.

Kaip vieną iš būdų įvertinti studentų teorines žinias naudosime teorinių klausimų testą. Testą sudarys 20 įvairaus tipo klausimų - pasirenkamieji, pasirenkamieji su keliais teisingais atsakymais ir laisvos atsakymo formos klausimai. Šis testas apima visas mokymo programos temas. Testo metu studentams gali reikėti įvertinti kodą ir nuspėti išvestį, surasti klaidą programiniame kode, savais žodžiais paaiškinti ir apibūdinti nurodytą programavimo konceptą arba parašyti trumpą funkciją atlikti duotai užduodžiai.

*Pastaba: Žemiau pateikti klausimai yra tik pavyzdinio tipo. Testams kurti naudojama kompiuterinė aplinka, jie nėra aprašomi šiame dokumente. Studentai klausimus gauna atsitiktine tvarka. Be to, kiekvienos grupės klausimai gali skirtis ir būti koreguojami grupės dėstytojo.*

Pavyzdiniai klausimai:

* 1. Kokia yra teisinga komanda norint sukurti naują „Create React App“ projektą (taip vadinamą „vieno puslapio aplikaciją” angl. „Single page application”)?
     1. npx create-react-app
     2. npx create-react-app myReactApp (+)
     3. npm create-react-app myReactApp
     4. npm create-react-app
  2. Kokia komanda naudojama React.js vietiniam kūrimo serveriui paleisti „Create React App“ aplinkoje?
     1. npm build
     2. npm run dev
     3. npm start (+)
     4. npm serve
  3. Norint sukurti ir paleisti React.js kodą, reikalingas Node.js:
     1. Tiesa (+)
     2. Netiesa
  4. Node.js veikia \_\_\_\_\_\_\_\_\_\_
     1. vartotojo sąsajoje (angl. „Client side”)
     2. serverio sąsajoje (angl. „Server side”) (+)
  5. Express.js yra \_\_\_\_\_\_\_\_\_\_
     1. node.js karkasas (+)
     2. React.js papildanti biblioteka
     3. programavimo kalba
  6. Kuris raktinis žodis sukuria konstantą „JavaScript“?
     1. var
     2. let
     3. constant
     4. const (+)
  7. React.js komponentų pavadinimai turi prasidėti didžiosiomis raidėmis.
     1. Tiesa (+)
     2. Netiesa
  8. Kokia yra teisinga „JavaScript“ sintaksė, norint pakeisti toliau esančio HTML elemento turinį?   
     <p id="demo">This is a demonstration.</p>
     1. #demo.innerHTML = "Hello World!"
     2. document.getElement("p").innerHTML = "Hello World!"
     3. document.getElementByName("p").innerHTML = "Hello World!"
     4. document.getElementById("demo").innerHTML = "Hello World!" (+)
  9. Kaip sukurti „JavaScript“ funkciją?
     1. function:myFunction()
     2. function=myFunction()
     3. function myFunction() (+)
  10. Kuri yra teisinga CSS sintaksė?
      1. {body:color=black;}
      2. {body;color:black;}
      3. body {color: black;} (+)
      4. body:color=black;
  11. Kuri CSS ypatybė valdo teksto dydį?
      1. font-size (+)
      2. font-style
      3. text-size
      4. text-style
  12. Kokia yra teisinga CSS sintaksė, kad visi <p> elementai būtų paryškinti?
      1. p {text-size:bold;}
      2. <p style="font-size:bold;">
      3. <p style="text-size:bold;">
      4. p {font-weight:bold;} (+)
  13. Naudojant užpildymo savybę (angl. „padding”) ar leidžiama naudoti neigiamas reikšmes?
      1. Taip
      2. Ne (+)
  14. Koks yra teisingas būdas aprašyti „JavaScript“ masyvą?
      1. const colors = "red", "green", "blue"
      2. const colors = ["red", "green", "blue"] (+)
      3. const colors = (1:"red", 2:"green", 3:"blue")
      4. const colors = 1 = ("red"), 2 = ("green"), 3 = ("blue")
  15. Kaip suapvalinti skaičių 7,25 iki artimiausio sveikojo skaičiaus?
      1. Math(7,25)
      2. Math.round(7,25) (+)
      3. Round(7,25)
      4. Math.change(7,25)
  16. Ką grąžins šis kodas: Boolean(10 > 9)
      1. NaN
      2. True (+)
      3. False
  17. Kuris operatorius gali būti naudojamas sąlyginiam React komponento atvaizdavimui?
      1. ||
      2. //
      3. $$
      4. && (+)
  18. Kai pateikiamas sąrašas React.js aplikacijoje naudojant JavaScript map() metodą, ko reikia kiekvienam pateikiamam elementui?
      1. index
      2. key (+)
      3. id
      4. name
  19. Kaip galite sujungti šiuos masyvus naudodami sklaidos (angl. „spread“) operatorių?

const array1 = [1, 2, 3];

const array2 = [4, 5, 6];

* + 1. const combined = ...array1 + ...array2
    2. const combined = array1 + array2
    3. const combined = [...array1, ...array2] (+)
    4. const combined = [array1, array2]
  1. Raskite klaidą šiame JSX kode:

function car({make, model}) {

return <h1>{make} {model}</h1>;

};

* + 1. Pirmoji funkcijos raidė turi būti didžioji (+)
    2. Būtina suvynioti „return“ į fragmentą
    3. Būtina pridėti skliaustus aplink „return“ reikšmę
    4. Reikėtų pašalinti funkcijos pavadinimą

**Baigiamasis projektas** *(vertė 90% galutinio balo)*. Trukmė: 15 val. 30 min. *Pateikiamos dvi galimos užduotys, studentams galima duoti vieną iš jų.*

Užduotis Nr. 1:

*Sukurkite React.js + Node.js + MongoDB programą, kuri leistų, renginių organizavimu užsiimančiai įmonei, registruoti telefonu paskambinusius asmenis į renginį.* Renginių organizatorius, prisijungęs prie programos, matys: a) Registracijos formą (joje nurodoma: i) vardas ir pavardė; ii) el.paštas; iii) amžius;) ir b) užsiregistravusių vartotojų sąrašą (nurodoma lentelė su vartotojo vardu ir pavarde, el. paštu, gimimo data, administravimo mygtukais (atnaujinti ir ištrinti vartotojus)).

Aplikacijos funkcionalumas:

1. CRUD operacijos (registracijos formoje galima sukurti vartotoją, lentelėje matyti, atnaujinti ir ištrinti vartotojus);
2. Duomenų saugojimas (vartotojai saugomi duomenų bazėje)
3. Dizainas. Dizainą sukurkite patys, tačiau minimalus dizainas būtinas.

Užduotis Nr. 2:

*Sukurkite React.js + Node.js + MongoDB programą, kuri leistų, grožio salonui leistų registruoti, telefonu paskambinusius klientus.* Grožio salonas, prisijungęs prie programos, matys: a) Registracijos formą (joje nurodoma: i) vardas ir pavardė; ii) el.paštas; iii) registracijos data ir laikas; ) ir b) užsiregistravusių vartotojų sąrašą (nurodoma lentelė su vartotojo vardu ir pavarde, el. paštu, registracijos data ir laiku, administravimo mygtukais (atnaujinti ir ištrinti vartotojus)).

Aplikacijos funkcionalumas:

1. CRUD operacijos (registracijos formoje galima sukurti vartotoją, lentelėje matyti, atnaujinti ir ištrinti vartotojus);
2. Duomenų saugojimas (vartotojai saugomi duomenų bazėje)
3. Dizainas. Dizainą sukurkite patys, tačiau minimalus dizainas būtinas.

Vertinimo kriterijai (tinka abiejų užduočių kontekste):

| **Kriterijus** | **Balo dalis** |
| --- | --- |
| *Kodo talpinimas ir versijavimas* | |
| Projekto patalpinimas [github.com](https://github.com/) (naudojamas GIT kodo versijamo sistema ir Github.com nuotolinė repozitorija sujungta su egzaminuojamo paskyra) platformoje | 1 |
| *Serverio pusė (“Server side”)* | |
| Node.js (express.js) serverio paleidimas ir Rest API konfigūracija | 1 |
| Taisyklingai sudaryti Rest API maršrutai (“routes/endpoints”) | 0.5 |
| Duomenų bazės (MongoDB) paleidimas, konfigūracija ir sujungimas su Node.js (express.js) serveriu | 1 |
| Taisyklingai sudaryti MongoDB duomenų modeliai | 0.5 |
| Užklausų apdorojimas ir klaidų sutvarkymas | 1 |
| *Kliento pusė (“Client side”)* | |
| React.js aplikacijos sukūrimas ir paleidimas | 0.5 |
| React.js aplikacijos sujungimas su serverio puse ir sukurtais Rest API maršrutais | 1 |
| Taisyklinga React.js aplikacijos medžio ir komponentų struktūra | 1 |
| CRUD operacijos (registracijos formoje galima sukurti vartotoją, lentelėje matyti, atnaujinti ir ištrinti vartotojus) | 2 |
| Logiškas ir suprantamas, bent minimalus stilius | 0.5 |

*Pastaba: Baigiamasis projektas taip pat gali skirtis kiekvienoje grupėje ir būti koreguojamas grupės dėstytojo. Taip norime sumažinti plagijavimo galimybes.*

1. Teorinės dalies testas atsiskaitomas paskutinės paskaitos metu (klausimus žr. 2 puslapyje). [↑](#footnote-ref-0)
2. Projektas kuriamas 4 paskutinių paskaitų metu, po teorinės dalies (projekto reikalavimai žr. 4 puslapyje). [↑](#footnote-ref-1)