

## Frontend – pažengusių (2 level)

### Programa:

#### 1. Front-end: HTML, CSS, Bootstrap karkasas (90 val.)

##### a. Potemės:

- i. *HTML pagrindai: Tagai, atributai, pagrindiniai elementai. Sąrašai ir lentelės. Paveikslukai ir video.*
- ii. *Puslapio struktūra: Puslapio struktūros elementai – header, footer, div.*
- iii. *HTML formos: Formos ir jų elementai*
- iv. *CSS pagrindai: Kas yra CSS, stilių panaudojimas ir baziniai pavyzdžiai, sintaksė.*
- v. *CSS tekstai, šriftai, spalvos: Darbas su teksto apipavidalinimu ir jo elementais.*
- vi. *Rėmai, pozicijos: Elementų pozicionavimas, margin/padding, rėmeliai, float.*
- vii. *Responsive design pagrindai: Kas yra responsive design, media queries ir mobilios versijos.*
- viii. *Twitter Bootstrap karkasas: Kas yra Twitter Bootstrap, jo klasės ir elementai, jo pagrindu sukurtų šablonų pavyzdžiai.*

#### 2. Front-end: JavaScript programavimo kalba (120 val.)

##### a. Potemės:

- i. *JavaScript sintaksė, stiliaus taisyklės*
- ii. *Kintamieji, duomenų tipai, operatoriai, sąlygos sakiniai, funkcijos, ciklai*
- iii. *DOM manipuliavimas, interaktyvumas*
- iv. *Klaidų valdymas (debugging, error handling)*
- v. *Objektinis programavimas su JavaScript*
- vi. *Paveldėjimas ir prototipai*
- vii. *Kreipimasis į serverį (JavaScript XML ir AJAX, JSONP)*
- viii. *Asinschroninis programavimas (promises, callbacks, listeners)*
- ix. *JavaScript APIs (canvas, geolocation, web workers, web sockets)*
- x. *JavaScript modulių sistemos (Common JS, AMD, ES2015)*
- xi. *JavaScript aplikacijos testavimas*
- xii. *Automatizavimas (Gulp, WebPack)*

#### 3. Backend-end: Node.js (Express.js), MongoDB (80 val.)

##### a. Potemės:

- i. *Node.js ir npm pagrindai*
- ii. *Express.js pagrindai*
- iii. *Rest API pagrindai*
- iv. *Express.js serverio kūrimas bei CRUD (Create, Read, Update, Delete) operacijos panaudojant Express.js*
- v. *Duomenų bazių pagrindai*

- vi. *Express.js serverio ir MongoDB duomenų bazės komunikavimas bei operacijos*
- vii. *Front-end ir Back-end (Express.js) sujungimas*
- viii. *"Server side rendering" ir "Client side rendering" pagrindai*

#### **4. Front-end: React.js karkasas (150 val.)**

##### **a. Potemės:**

##### **i. Pagrindiniai React.js karkaso konceptai:**

1. *JSX*
2. *Class Components*
  - a. *setState and component lifecycle methods*
3. *Function Components*
  - a. *hooks: useState and useEffect, useReducer, useRef, useMemo, useCallback, custom hooks*
4. *Props and State*
5. *Conditional Rendering*
6. *Lists and keys*
7. *Forms*
8. *Context*
9. *Higher Order Components*
10. *Render props*
11. *Refs*
12. *Error Boundaries*
13. *Portals*
14. *HTTP Requests*
15. *State Management — Redux, Recoil, React Query*
16. *Routing for "Create React App" - React Router*

##### **ii. React.js aplikacijos testavimas**

##### **iii. Karkasai: "Create React App", (Next.js, Gatsby.js, Remix – pagrindai)**

##### **iv. Programavimo aplinkos ir kūrimo proceso valdymas ir versijų kontrolė.**