

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**T-PLANNER – Aplicație pentru gestionarea
trainingurilor pentru studenți**

propusă de

Bianca-Adelina Dascălu

Sesiunea: *iulie, 2019*

Coordonator științific

Conf. Dr. Anca Vitcu

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

T-PLANNER – Aplicație pentru gestionarea trainingurilor pentru studenți

Bianca-Adelina Dascălu

Sesiunea: *iulie, 2019*

Coordonator științific

Conf. Dr. Anca Vitcu

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP,
absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de
..... specializarea, promoția
....., declar pe propria răspundere, cunoscând consecințele falsului în
declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr.
1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elaborată sub îndrumarea dl. / d-na
_____, pe care urmează să o susțină în fața
comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin
orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea
conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări
științifice în vederea facilitării fașificării de către cumpărător a calității de autor al unei
lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere
că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*T-Planner – aplicație pentru gestionarea trainingurilor pentru studenți*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 27.06.2019

Absolvent *Bianca-Adelina Dascălu*

(semnătura în original)

Cuprins

Introducere	4
Context.....	4
Motivație.....	4
Descrierea soluției	4
Abordare Tehnică.....	5
Structura lucrării	6
Contribuții	7
1 Capitolul 1 – Arhitectura aplicației și abordare tehnică	8
1.1 Arhitectura aplicației.....	8
1.1.1 React	9
1.1.2 MobX.....	9
1.1.3 Firebase.....	10
1.2 Structura folderelor pe partea de client	11
2 Capitolul 2- Rolul de trainer	13
2.1 Descrierea problemei.....	13
2.2 Abordări anterioare	13
2.3 Descrierea soluției.....	13
2.3.1 Funcționalități specifice cursului.....	14
2.3.2 Detalii de implementare	18
3 Capitolul 3 – Rolul de participant.....	23
3.1 Descrierea problemei.....	23
3.2 Descrierea soluției.....	23
3.2.1 Funcționalități specifice utilizatorului cu rol de participant	23
3.2.2 Detalii de implementare	24
4 Capitolul 4- Rolul de admin.....	26
4.1 Descrierea problemei.....	26
4.2 Descrierea soluției.....	26
4.2.1 Funcționalități pentru utilizatorii cu rolul de admin	26
4.2.2 Detalii de implementare	29
Concluzii și direcții viitoare	30
Bibliografie	31

Introducere

În această secțiune este prezentată pe scurt tema aplicației “T-Planner”, contextul în care se află, soluția, precum și tehnologiile folosite în implementarea acesteia.

Context

Studentii Facultății de Informatică din Iași organizează în fiecare an un proiect foarte cunoscut în comunitatea IT locală, „FII Practic”, proiect care aduce beneficii atât studenților care se înscriu la traininguri, cât și companiilor IT din Iași care se implică în mod activ prin susținerea trainingurilor pentru studenți. Am avut ocazia să cunosc acest proiect ca și participant la cursuri, ca și membru din organizare și staff și, nu în ultimul rând, ca și trainer.

În aplicația actuală “FII Practic”, studenții își fac un cont, vizualizează cursurile, aplică, după care sunt sunați pentru confirmare, iar un trainer poate vizualiza lista de aplicanți și să-i selecteze pe cei care se potrivesc cerințelor.

Motivație

Aplicația „T-Planner” urmărește să ofere un mediu dinamic și flexibil pentru organizarea trainingurilor pentru studenți.

Motivația a pornit în momentul în care am fost trainer, iar toate informațiile pe care noi, trainerii, trebuia să le oferim organizatorilor pentru a apărea pe site trebuiau să treacă prin mai multe persoane și prin mai multe căi de comunicare. Cu toate acestea, au existat momente în care informațiile au ajuns cu anumite erori pe site. De asemenea, aplicația actuală trebuie refăcută în mare parte de la an la an de către studenți, iar din cauza faptului că perioada în care începe proiectul coincide cu perioada sesiunii pentru studenți apar foarte multe întârzieri care, din ce am observat, fac companiile de IT din Iași să-și piardă încrederea în studenții organizatori și în calitatea proiectului.

Descrierea soluției

Prin aplicația T-Planner, admin-ul poate crea dinamic ariile precum și cursurile ediției respective, promovează membri pentru a avea rolul de trainer sau admin, asignează traineri la

cursuri, primește lista cu membrii acceptați care vor fi ulterior contactați de către organizatori, vizualizează rapoarte de prezență pentru sesiunile de curs precum și feedback-ul primit de la utilizatori, fie ei participanți sau traineri. Ca și trainer, cursul pe care îl susții are o secțiune proprie din care poți seta întrebările pentru formularul de aplicare, poți vizualiza aplicații, membrii, poți face prezența la curs într-un mod rapid și modern și poți vizualiza feedback-ul studenților participanți.

Abordare Tehnică

Aplicația T-Planner este o aplicație web de tip *server-less*, iar în implementarea acesteia s-au folosit următoarele tehnologii:

- **REACT** – am folosit această bibliotecă de JavaScript întrucât este mai rapidă și se diferențiază prin faptul că se folosește de Virtual DOM (Document Object Model). Asta înseamnă că dacă doar o componentă din pagină necesită o actualizare, nu se va randa toată pagina ci doar acea componentă.
- **Firebase** – am folosit această platformă pentru partea de *backend-as-a-service* și baza de date întrucât oferă o bază de date *real-time* care m-a ajutat în implementarea funcționalității de prezență la cursuri, actualizarea informațiilor în urma unor acțiuni și pentru stocarea datelor dinamic.
- **MobX** – am folosit această bibliotecă pentru a administra *state*-ul aplicației într-un singur loc pentru componentele care necesită niște proprietăți comune. React și MobX se îmbină foarte bine, MobX furnizând mecanisme de stocare și actualizare a *state*-ului aplicației pe care React o randează.
- **MaterialUI** – am folosit această bibliotecă pentru a mă ajuta la a crea interfața aplicației deoarece oferă tot felul de componente reutilizabile și ușor customizabile pentru a crea aplicații web.

Structura lucrării

Prezenta lucrare este alcătuită dintr-o secțiune de contribuții personale, urmată de 4 capitole principale, primul evidențiind detalii despre arhitectura aplicației, iar următoarele 3 axându-se pe funcționalitățile implementate pentru fiecare tip de utilizator, acestea fiind:

- Rolul de admin;
- Rolul de trainer;
- Rolul de participant;

încheindu-se cu o secțiune de concluzii generale și direcții viitoare.

Contribuții

În timpul implementării aplicației am pus accentul pe crearea unui mediu cât mai dinamic pentru organizarea trainingurilor. Obiectivul principal fiind extinderea funcționalității aplicației actuale “FII Practic” astfel încât trainerii din companiile IT din Iași cât și organizatorii proiectului (care se schimbă de la an la an) să găsească un mediu plăcut și ușor de folosit în care să aibă toate resursele necesare pentru buna funcționare și organizare în momentul începerii proiectului, am reușit să realizez următoarele:

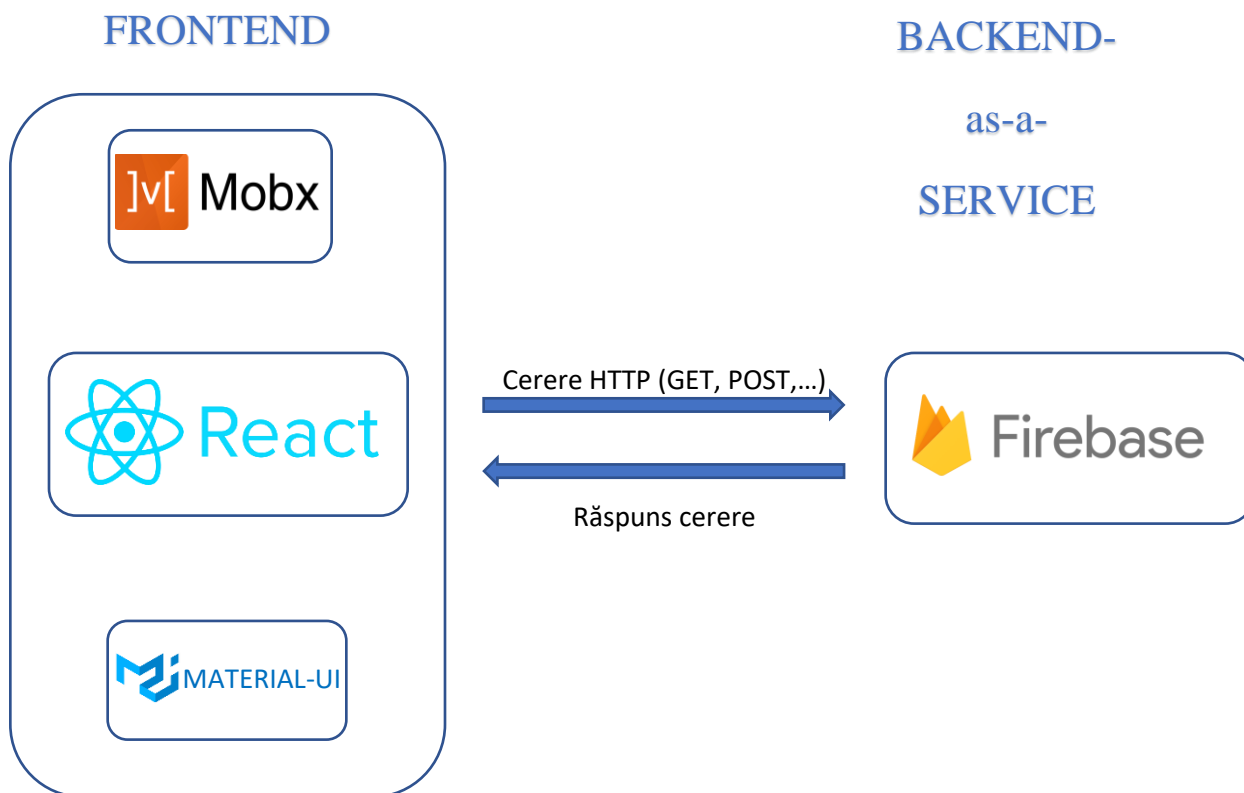
- Crearea unui mediu stabil și dinamic pentru stocarea și editarea informațiilor despre trainingurile disponibile;
- Vizualizarea în timp real a modificărilor făcute de către admin și/sau trainer în cadrul unor funcționalități;
- Posibilitatea creării listelor de prezență folosind tehnologii precum scanarea de coduri QR;
- Crearea de *pie-chart*-uri pentru vizualizarea raportului dintre membrii prezenți și totalul de membri de la un curs;
- Posibilitatea trimiterii feedback-ului și vizualizării acestuia direct din aplicație;
- Implementarea unui algoritm de calculare a frecvenței cuvintelor pentru a extrage cele mai importante informații din feedback;
- Salvarea eficientă a datelor despre cursuri necesare utilizatorilor.

1 Capitolul 1 – Arhitectura aplicației și abordare tehnică

În acest capitol voi prezenta detalii despre arhitectura și structura aplicației precum și detalii despre abordarea tehnică a implementării soluției.

1.1 Arhitectura aplicației

Aplicația T-Planner este o aplicație web implementată cu Javascript, HTML și CSS (prin biblioteca React), pe partea de client, și Firebase pe partea de *backend*. Are o arhitectură de tip *server-less* ceea ce înseamnă că nu am folosit niciun server pe partea de *backend* ci am folosit o platformă de tipul *Backend-as-a-service (BAAS)*, anume Firebase, care practic furnizează un API prin care poți trimite și salva date, precum și să autentifici utilizatori.



1.1.1 React

React este o bibliotecă de JavaScript creată și menținută de Facebook care se folosește de sintaxa JSX pentru a descrie cum ar trebui să arate UI-ul (*User Interface*). Față de elementele din DOM-ul (*Document Object Model*) din *browser*, elementele din React sunt obiecte ușor de creat. DOM-ul din React se ocupă de actualizarea DOM-ului din *browser* având grijă să coincidă cu elementele din React. Conceptul de *state* din React se referă la starea unei anumite componente la un moment dat. În *state* ținem de obicei datele care se pot modifica în timpul utilizării aplicației și care duc la schimbarea UI-ului, fapt care se rezolvă apelând metoda *setState* din React, aceasta cauzând rerandarea componentei care folosește *state*-ul respectiv. Conceptul de *props* reprezintă de fapt date primite ca input de o componentă de la componenta părinte. Prin intermediul *props*-urilor conectăm componentele de tip părinte-copil între ele.

1.1.2 MobX

MobX este o bibliotecă care furnizează mecanisme pentru stocarea și actualizarea stării aplicației prin intermediul unor *store*-uri. Rolul acestora este de a muta logica și starea aplicației înafara componentelor într-o unitate independentă care astfel poate fi folosită în restul aplicației. Spre exemplu, în momentul autentificării unui utilizator, voi seta în *store*-ul pentru Sesiune utilizatorul autentificat (Fig. 1 – Setare utilizator autentificat în store), astfel mă pot folosi de proprietățile acestuia în orice componentă din aplicație.

```
class SessionStore {  
  @observable  
  authUser = null;  
  
  //setez global în aplicație utilizatorul autentificat  
  @action  
  setAuthUser = authUser => {  
    this.authUser = authUser;  
  };  
}  
  
export default new SessionStore();
```

Fig. 1 – Setare utilizator autentificat în store

Am ales MobX pentru a administra *state*-ul aplicației deoarece este o soluție simplă, scalabilă și, spre deosebire de alte soluții de gestionare a *state*-ului (precum Redux), aceasta nu restricționează modul prin care poți modifica *state*-ul, cum ar fi să-l facă imutabil.

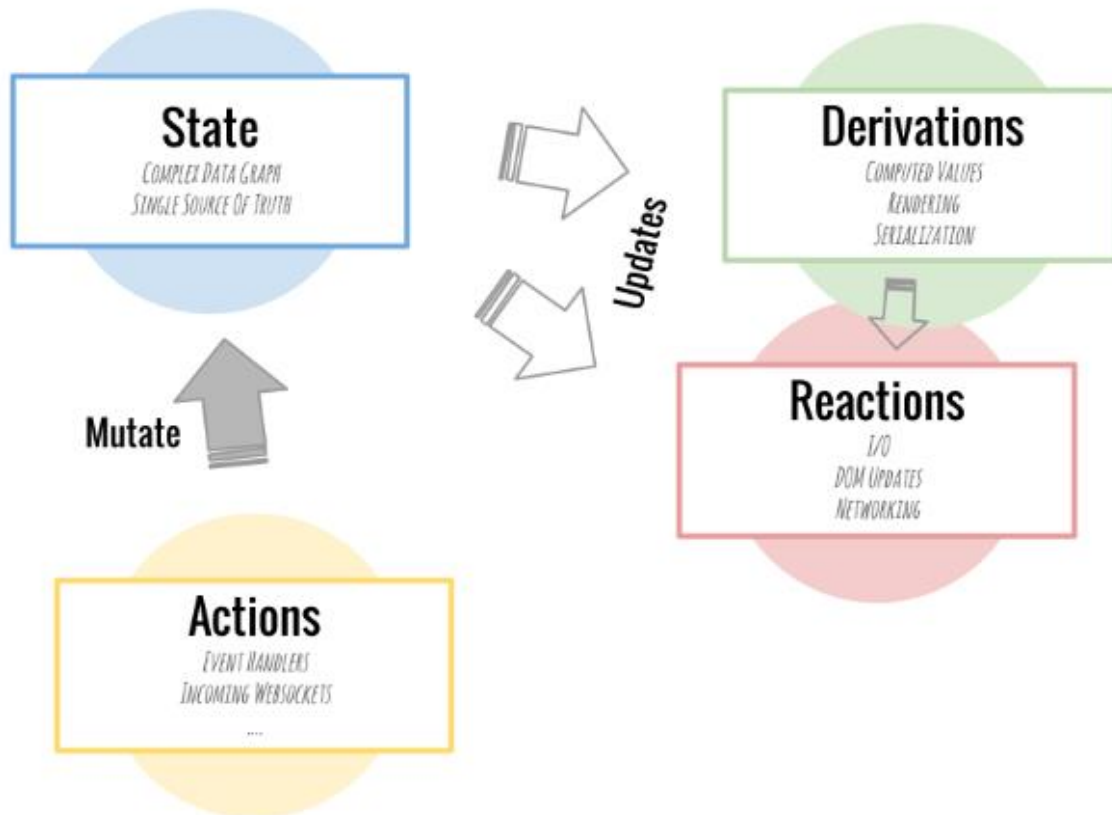


Fig. 2 – Cum funcționează MobX (preluată din documentația oficială MobX)

1.1.3 Firebase

Firebase este o platformă de tip *Backend-as-a-service* pentru dezvoltarea aplicațiilor care oferă servicii backend precum baze de date *real-time*, stocare *cloud*, autentificare, învățare automată și altele. Principalele avantaje ale Firebase sunt complexitatea redusă, viteza de dezvoltare, scalabilitate, securitate.

1.2 Structura folderelor pe partea de client

Am structurat proiectul (Fig. 3) în primul rând în funcție de componentele mari. Astfel observăm *Authentication* pentru Login și Creare cont nou, *MainPage* pentru toate componentele de pe pagina principală, *NavigationBar* pentru bara de navigare care este prezentă în toate paginile și *Dashboard* unde am împărțit în alte trei subdirectoare în funcție de rolul utilizatorului. Acolo am adăugat toate componentele ce țin strict de pagina de cont a fiecărui tip de utilizator. În directorul *Firebase* am centralizat fișierele de configurare și metodele folosite cu Firebase, iar în directorul *Forms* am adăugat toate formularele implementate în aplicație.

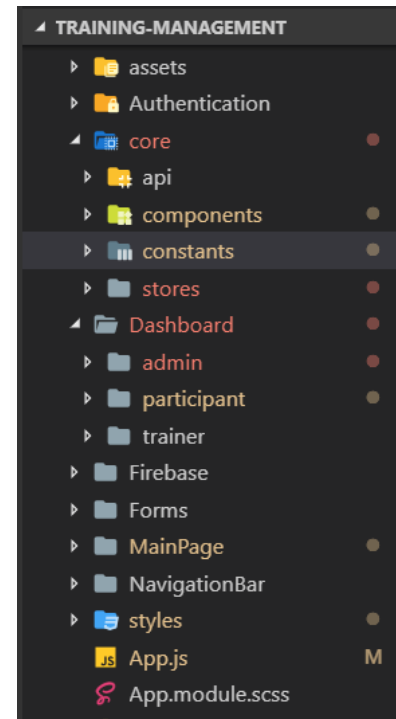


Fig. 3 – Structura fișierelor

În directorul *core* am adăugat constantele folosite în aplicație, mai exact rutele, *store*-urile pentru MobX, unele componente reutilizabile, precum și fișierele cu *request*-urile la baza de date de citire, scriere, actualizare și ștergere (Exemplu de astfel de fișier pentru cursuri în Fig. 4).

```

export const getCourses = async () => {
  return axios.get(`${baseUrl}.json`);
};

export const getCourse = async (courseId) => {
  return axios.get(`${baseUrl}/${courseId}.json`);
};

export const addCourse = async (courseInfo) => {
  return axios.post(`${baseUrl}.json`, courseInfo);
};

export const updateCourse = async (updatedCourse) => {
  return axios.put(`${baseUrl}/${updatedCourse.courseId}.json`, { ...updatedCourse, courseId: updatedCourse.courseId });
};

export const deleteCourse = async (courseId) => {
  return axios.delete(`${baseUrl}/${courseId}.json`);
};

```

Fig. 4 – Request-uri HTTP la baza de date din Firebase pentru cursuri

Toate componentele sunt formate din două tipuri de fișiere, anume *NumeComponentă.js* și *NumeComponentă.module.scss*. Cele cu extensia „.js” reprezintă fișierul JavaScript cu toată logica componentei precum și cu partea de JSX care reprezintă *template*-ul componentei, iar cele cu extensia „.module.scss” sunt fișiere SasS (preprocesor de CSS) și conțin stilurile specifice componentei.

În directorul *styles* sunt fișierele cu variabilele globale folosite pentru stiluri precum și *mixin*-uri de stiluri pentru a nu repeta în cod anumite stiluri refolosite și pentru a ajuta la *responsiveness*.

2 Capitolul 2- Rolul de trainer

În acest capitol voi prezenta funcționalitățile pe care le oferă aplicația pentru utilizatorii cu rolul de trainer.

2.1 Descrierea problemei

Ca trainer, am întâmpinat probleme la trimiterea formularului de aplicare, deoarece a avut traseul: trainer – HR – organizator FII Practic – Departament IT – Site FII Practic, și tot a ajuns cu niște erori. De asemenea, noi nu aveam acces la lista de prezență de la cursuri, iar orice alte chestiuni care trebuiau comunicate se transmiteau prin tot felul de platforme precum Slack sau Google Forms.

2.2 Abordări anterioare

În aplicația actuală, un trainer poate vedea lista cu aplicanți și răspunsurile lor la formularul de aplicare, să-i accepte pe cei care îndeplinesc criteriile de selecție și să vizualizeze lista finală de membri în urma confirmărilor.

2.3 Descrierea soluției

Un trainer își va face cont în aplicație după care va aștepta până un admin îi va oferi rolul de trainer și îl va asigna unui curs. Când toate acestea s-au finalizat, trainerul are acces la o pagină separată cu detaliile contului său, o listă cu toate cursurile la care a fost asignat, precum și un formular de feedback pentru organizatorii proiectului (Fig. 5).

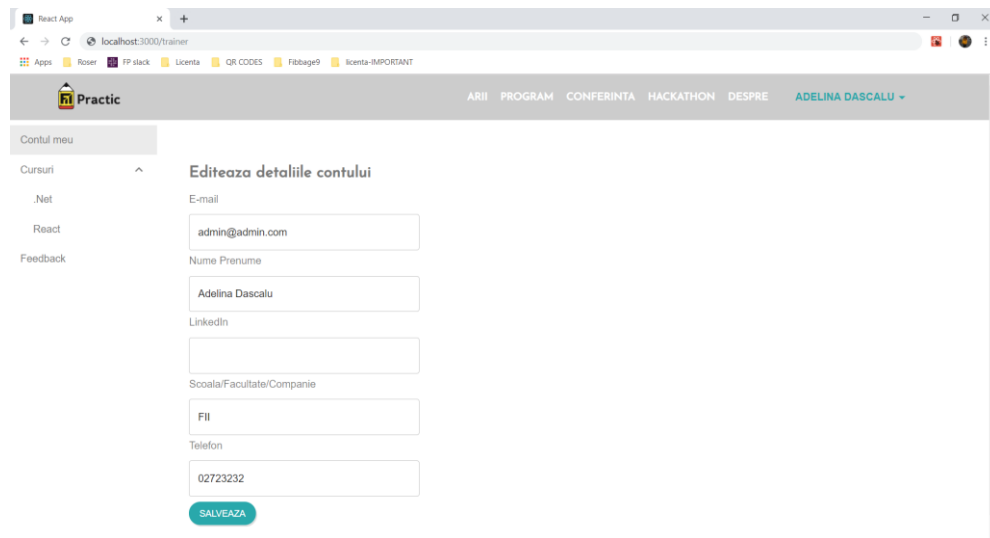


Fig. 5 – Contul de trainer

2.3.1 Funcționalități specifice cursului

Pentru fiecare curs la care a fost asignat, un trainer va putea seta direct din aplicație întrebările pentru **formularul de aplicare** (Fig. 6), putând să adauge oricâte întrebări sunt necesare. Astfel, formularul va apărea direct în aplicație în momentul în care cineva va dori să aplice la cursul respectiv, nemaifiind nevoie să fie trimis organizatorilor și să fie adăugat de admin.

REACT

FORMULAR DE APLICARE
APLICANTI
MEMBRI
PREZENTE
ANUNTURI
FEEDBACK STUDENTI

Adauga intrebarile pentru formularul de aplicare. Cand ai terminat salveaza modificarile. Poti modifica intrebarile mai tarziu.

Intrebare

Care este specificitatea selectorilor in css?

Intrebare

Ce stiti de conceptul de state?

Intrebare

Ce stiti despre conceptul de props?

Intrebare

Ati mai lucrat cu React?

Intrebare

Care este structura unui HTML?

+
SALVEAZA

Fig. 6 – Setare formular de aplicare pentru un anumit curs

Trainerul are acces la lista cu aplicanți (Fig. 7) precum și la răspunsurile acestora la formularul de aplicare, putând astfel să-i accepte pe cei care îndeplinesc criteriile. Cei acceptați vor apărea într-o altă secțiune, “Membri” (Fig. 8), unde trainerul va putea vedea dacă utilizatorul respectiv a confirmat prezența sa la curs sau nu.

REACT

FORMULAR DE APLICARE **APLICANTI** MEMBRI PREZENTE

Aplicant	Formular de aplicare	Data	Actiuni
aplicant#0	VEZI RASPUNSURI	12.02.2020	<button>STERGE</button>
aplicant#1	VEZI RASPUNSURI	12.02.2020	<button>STERGE</button>
aplicant#2	VEZI RASPUNSURI	23/06/2019	<button>ACCEPTA</button>
aplicant#3	VEZI RASPUNSURI	12.02.2020	<button>STERGE</button>

Fig. 7 - Aplicanți

REACT

FORMULAR DE APLICARE APLICANTI **MEMBRI** PREZENTE

Nume	Formular de aplicare	Data	Confirmed
Andrei Pi	VEZI RASPUNSURI	12.02.2020	✗
Adelina Dascalu	VEZI RASPUNSURI	12.02.2020	✓
Stefania Matei	VEZI RASPUNSURI	12.02.2020	✓

Fig. 8 – Membri

Următoarea secțiune importantă este prezența la curs. Până în prezent, prezența la cursuri este făcută manual cu ajutorul stundenților voluntari care stau în fața sălii de training și strâng semnături de la toți cei care vin. Cu aplicația T-Planner, prezența la cursuri se realizează direct din aplicație astfel:

În timpul unui curs, trainerul va intra în secțiunea de prezențe și va deschide o nouă sesiune de prezență. Aceasta se creează cu data cursului, numărul maxim de membri care sunt în sală la momentul respectiv și un cod unic ales de trainer (Fig. 9).

Adauga o sesiune de prezenta noua

Data

06/12/2020

Numarul maxim de membri

10

Codul unic al sesiunii

.....

SUBMIT

Fig. 9 Creare sesiune de prezență

După ce s-a creat sesiunea, codul unic adăugat de trainer va genera un QR Code (*Quick Response Code*) care va rămâne disponibil până când trainerul va închide sesiunea. Codul va putea fi pus pe proiectorul video din sala de curs, iar participanții vor putea scana acel QR Code din contul lor din aplicația T-Planner și vor fi automat trecuți pe lista de prezență care va fi generată în timp real pe ecran (Fig. 10). Ca alternativă, în cazul în care sunt participanți în sală care nu au un *smartphone* la dispoziție, trainerul va putea comunica acel cod unic scris la crearea sesiunii, iar participanții se vor putea înscrie pe lista de prezență cu acel cod.



Fig. 10 – Sesiune de prezență activă

De asemenea, în timp ce studenții prezenți scanează codul QR și apar în lista de prezență, în această secțiune se poate observa un *pie-chart* care prezintă un raport statistic între studenții

prezenți la acea sesiune de curs și toți membrii participanți la acel curs. Acest raport se actualizează în timp real odată cu lista de membri (Fig. 11).

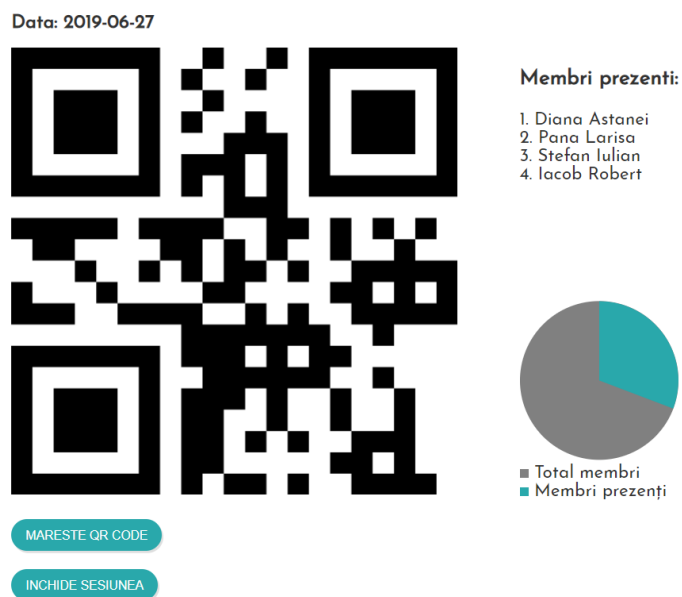


Fig. 11 – Pie-chart reprezentând raportul membri prezenți/total membri

Când lista de prezență este completă, trainerul va închide sesiunea de prezență și astfel nimeni nu o va mai putea accesa.

Ultima funcționalitate specifică cursului pe partea de trainer este secțiunea de feedback în care trainerul va putea vedea în mod anonim mesajele de feedback oferite de studenții participanți.

Aceste mesaje sunt precedate de un *word cloud* format pe baza frecvenței celor mai importante și definitorii cuvinte din mesajele de feedback de la studenți (Fig. 12).

Membrii și aplicanții sunt salvați în baza de date a unui curs ca un array de Id-uri ale utilizatorilor respectivi. În funcție de acele Id-uri sunt afișate listele cu membrii și aplicanții și proprietățile lor specifice.

În secțiunea de prezențe, am folosit componenta `qrcode.react` pentru a genera codul QR în funcție de codul unic adăugat de trainer. Aceasta este practic o componentă care primește ca input un string și afișează codul QR corespunzător sub forma unui `svg`. Pentru a genera lista de membri în timp real am creat o referință în `firebase` (Fig. 14 – Referință pentru membrii prezenți) care primește ca input Id-ul cursului și Id-ul sesiunii de prezență, și afișează datele corespunzătoare. Astfel, oricând se fac modificări în acel nod al bazei de date, se va actualiza și componenta creată în React care afișează acele date.

```
attendees = (courseId, sessionId) => (  
  
  this.db.ref(`courses/${courseId}/attendance/${sessionId}/attendees`);  
  
)
```

Fig. 14 – Referință pentru membrii prezenți

Pie-chart-ul de la secțiunea de prezență este un element de tip `svg` creat automat pe baza unor atribute specifice și a unor puncte bine definite (Fig. 15).

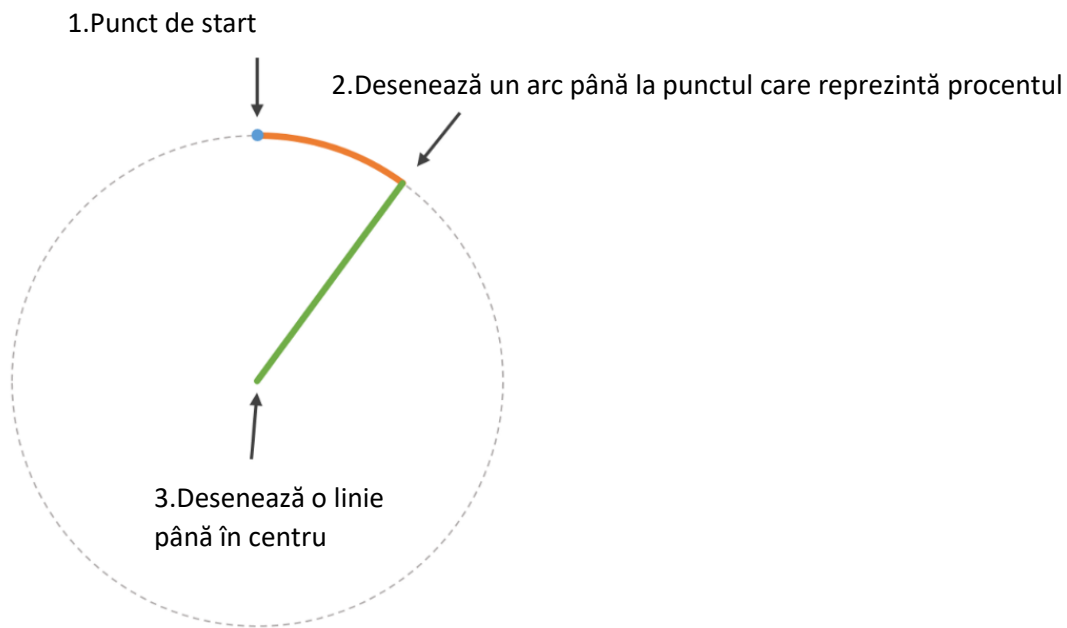


Fig. 15 – Modul de desenare a unei 'felii' dintr-un *pie-chart*

Pentru crearea *pie-chart*-ului, am început mereu de la poziția (0, 0) pe axa xOy, iar la fiecare actualizare a listei de prezență am calculat procentajul membrilor prezenți din totalul de membri. În funcție de procentul aflat, am calculat coordonatele celui de-al doilea punct de pe arcul cercului astfel:

```
getCoordinatesByPercent = (percent) => {
  const x = Math.cos(2 * Math.PI * percent);
  const y = Math.sin(2 * Math.PI * percent);
  return [x, y];
}
```

Desenarea efectivă a *pie-chart*-ului am implementat-o printr-o funcție (*displaySlices*) care calculează coordonatele celor două puncte definitorii ale arcului de cerc reprezentativ pentru procentaj după care formează atributul *d* (*data*) al elementului *path* care creează svg-ul. *M*, *A* și *L* sunt cei 3 pași menționați în Fig. 15 de desenare a unei „felii” dintr-un *pie-chart*.

```

displaySlices = (slices) => {
  let cumulativePercent = 0;
  const mappedSlices = slices.map(slice => {
    const [startX, startY] = this.getCoordinatesByPercent(cumulativePercent);
    const [endX, endY] = this.getCoordinatesByPercent(cumulativePercent + slice.percent);
    const largeArcFlag = slice.percent >= .5 ? 1 : 0;
    slice.pathData = [
      `M ${startX} ${startY}`,
      `A 1 1 0 ${largeArcFlag} 1 ${endX} ${endY}`,
      `L 0 0`
    ];
    cumulativePercent = cumulativePercent + slice.percent;
    return slice;
  });
  return (
    <path key={slice.value} d={slice.pathData} fill={slice.color} />
  )
}

```

Table 1 – Algoritm de creare a unui pie-chart format svg

Pentru calcularea frecvenței cuvintelor din mesajele de feedback, mai întâi am eliminat toate cuvintele care conțin cratime folosind un șablon de expresii regulate considerând că acele cuvinte din limba română nu sunt definatorii pentru feedback. După care, tot folosindu-mă de expresii regulate am eliminat toate caracterele speciale folosite în mesaje, precum semnele de punctuație. De asemenea, am folosit în formarea *word cloud*-ului doar cuvintele care erau formate din mai mult de 4 litere deoarece am considerat că sunt mai multe șanse ca acestea să fie cuvinte de

legătură, prepoziții sau alte cuvinte care nu aduceau un plus de valoare. Toate acestea fiind făcute, am calculat numărul aparițiilor acestor cuvinte, iar pentru ca *word-cloud*-ul să nu conțină discrepanță prea mare în dimensiunea cuvintelor, le-am sortat în ordine crescătoare în funcție de numărul de apariții și m-am folosit de indexul acestora în listă pentru calcularea dimensiunii lor.

```
getFrequency = (text) => {
  const dashedWordsPattern = /(?\S*['-])([a-zA-Z'-]+)/g;
  const textWithoutDashedWords = text.replace(dashedWordsPattern, "");

  const specialCharsPattern = /[\\.,-\/#!$%^&\\*;;:{}=\\_`~()\\?\\~]/g;
  const cleanText = textWithoutDashedWords.replace(specialCharsPattern, "");

  const allWords = cleanText.split(' ');

  const importantWords = allWords.filter(word => word.length >= 5);

  const frequencies = {};

  importantWords.forEach(word => {
    frequencies[word] = frequencies[word] ? frequencies[word] + 1 : 1;
  })

  const frequenciesArray = Object.keys(frequencies)
    .sort((word1, word2) => frequencies[word1] - frequencies[word2])
    .map((word, index) => {
      return { text: word, value: (index + 1) }
    })
  return frequenciesArray;
}
```


3 Capitolul 3 – Rolul de participant

În acest capitol voi prezenta funcționalitățile implementate pentru un utilizator care are rolul de participant la curs.

3.1 Descrierea problemei

În aplicația actuală, un utilizator poate vizualiza informații despre ariile, și cursurile corespunzătoare precum și evenimentele conexe proiectului. Acesta poate aplica la unul sau mai multe cursuri completând formularul de înscriere și să aștepte până va fi contactat pentru confirmare, în cazul în care este acceptat la unul din cursuri.

3.2 Descrierea soluției

În aplicația T-Planner, în momentul în care un utilizator este trecut în baza de date ca și cum a confirmat prezența sa la un curs la care a fost acceptat, în secțiunea “Contul meu” îi apare cursul (sau cursurile) respectiv/e.

3.2.1 Funcționalități specifice utilizatorului cu rol de participant

Participantul poate vedea detaliile cursului și programa stabilită de traineri. De asemenea, din această secțiune își poate valida prezența la curs.

Astfel, dacă există o sesiune activă de prezență, utilizatorul are opțiunea să scaneze un cod QR (afișat de trainer cu proiectorul video în timpul cursului) sau să introducă codul unic al sesiunii de prezență (Fig. 16).

React

DETALII PREZENTE ANUNTURI FEEDBACK

Pentru a fi trecut ca *prezent* la aceasta sesiune, scanati QRcode-ul dat de trainer la curs SAU introduceti codul unic al sesiunii (comunicat de catre trainer).

SCAN QR CODE

SAU

Introdu codul unic al sesiunii

SUBMIT

Fig. 16 – Sesiune activă de prezență

O altă secțiune vizibilă pentru participant în cadrul cursului, este cea de feedback unde utilizatorul poate să comunice părerile sale constructive asupra trainingului respectiv, urmând a fi vizualizate de către trainerii.

Separat de curs, sunt secțiunile de editare a datelor contului, precum, nume, e-mail, școală/facultate, număr de telefon, dar și secțiunea de feedback pentru întregul proiect și organizatori, nemaifiind nevoie ca organizatorii să trimită formularul de feedback pe alte căi de comunicare.

3.2.2 Detalii de implementare

Pentru implementarea funcționalității de înscriere în lista de prezență a unui participant am folosit componenta React **react-qrcode-reader** pentru citirea codurilor QR prin intermediul camerei web. Aceasta cere accesul la camera web a dispozitivului folosit, detectează coduri QR și returnează codul unic al sesiunii de prezență în cazul în care scanarea s-a efectuat cu succes. Pe partea de client verific dacă string-ul returnat de această componentă este identic cu codul unic al sesiunii de prezență active, iar în caz afirmativ, utilizatorul este trecut cu succes în lista de prezență și afișat în timp real pe pagina de curs a trainerului.

```

handleScan = (data) => {

  if (data) {

    const { handleAttendance, activeSession, sessionStore: { authUser } } = this.props;

    // verific dacă codul unic al sesiunii este identic cu cel returnat de codul QR

    if (activeSession.uniqueCode === data) {

      //trec utilizatorul pe lista de prezență

      handleAttendance(authUser);

    } else {

      this.handleWrongCode();

    }

    // închide camera web

    this.setState({ scan: false });

  }
}

```

Fig. 17 – Funcție de scanare a unui cod QR

4 Capitolul 4- Rolul de admin

În acest capitol voi prezenta funcționalitățile pe care le oferă aplicația T-Planner pentru rolul de admin.

4.1 Descrierea problemei

În fiecare an un nou grup de studenți trebuie să actualizeze aplicația FII Practic, căutând în cod locațiile unde sunt trecute date precum detaliile despre cursuri. Acești studenți sunt de multe ori începători și au nevoie de mai multe cunoștințe pentru a înțelege și a modifica acel cod. Acest lucru nu ar fi o problemă dacă nu ar exista date limită în cadrul acestui proiect și dacă acea perioadă nu ar coincide cu sesiunea de examene de la facultate.

4.2 Descrierea soluției

În aplicația T-Planner adminul poate edita utilizatorii, adăuga arii și respectiv cursuri, administra aplicanții, vizualiza listele de prezență precum și feedback-ul primit prin intermediul aplicației de la participanți și traineri.

4.2.1 Funcționalități pentru utilizatorii cu rolul de admin

Adminul vizualizând lista cu toți utilizatorii (Fig. 18), în momentul în care se stabilesc trainerii de la companii, acesta le poate oferi lor rolul de *trainer* în aplicație. De asemenea poate oferi și rolul de *admin* altor utilizatori.

9 users

Nume	E-mail	Roluri	Actiuni
Diana Astanei	d.astanei@trainer.com		PROMOVEAZA LA ADMIN STERGE TRAINER
Andrei Pi	andreipi@user.com		PROMOVEAZA LA TRAINER PROMOVEAZA LA ADMIN
Adelina Dascalu	a.dascalu@trainer.com	TRAINER	PROMOVEAZA LA ADMIN STERGE TRAINER
Adelina Admin	ade.admin@admin.com	ADMIN	PROMOVEAZA LA TRAINER STERGE ADMIN
Ovidiu Grigoras	o.grigoras@trainer.com		PROMOVEAZA LA ADMIN STERGE TRAINER
Stefan Trupina	s.trupina@trainer.com	TRAINER	PROMOVEAZA LA ADMIN STERGE TRAINER
Adelina Dascalu	admin@admin.com	TRAINER ADMIN	STERGE TRAINER STERGE ADMIN
Stefania Matei	s.matei@trainer.com	TRAINER	PROMOVEAZA LA ADMIN STERGE TRAINER
Trainer Account	trainer@trainer.com	TRAINER	PROMOVEAZA LA ADMIN STERGE TRAINER

Fig. 18 – Lista utilizatorilor

Adăugarea unei noi arii sau editarea uneia deja existente se realizează rapid și ușor, doar prin completarea unor formulare cu detaliile necesare (Fig. 19). Cursurile se creează (sau editează) în același loc cu informațiile corespunzătoare, putând chiar să asignezi traineri specifici. Când un trainer a fost asignat unui curs, acestuia îi apare în contul său secțiunea pentru cursul respectiv cu toate funcționalitățile menționate în Capitolul 2- Rolul de trainer.

Fig. 19 – Secțiune de editare a unei arii și a cursurilor specifice

Adminul are și lista cu toți aplicanții acceptați de trainerii, email-urile lor precum și numărul lor de telefon, listă specifică fiecărui curs. În momentul în care organizatorii contactează aplicanții, aceștia au posibilitatea direct din aplicație să-i diferențieze pe cei care au confirmat bifând dacă au fost contactați, iar în cazul în care au și confirmat, bifând acest lucru de asemenea.

React				
Nume	E-mail	Telefon	Contactat	Confirmat
Stefania Matei	s.matei@trainer.com	0723723232	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Andrei Pi	andreipi@user.com	0722723232	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Adelina Dascalu	admin@admin.com	0722723232	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 20 – Listă de aplicanți acceptați de trainerii pentru un anumit curs

De asemenea, adminul are acces la toate listele de prezență specifice pentru fiecare sesiune a fiecărui curs. În funcție de aceasta, organizatorii fac diplomele și premiile din partea proiectului.

React	
2020-06-06	
Stefan Andrei	
Adelina Dascalu	
2020-06-12	
Stefania Matei	

Fig. 21 – Liste de prezență specifice sesiunilor unui anumit curs

Nu în ultimul rând, adminul poate vizualiza din aplicație tot feedback-ul primit în mod anonim din partea trainerilor și a participanților precedat de asemenea de un *word-cloud*, implementat la fel ca cel de la secțiunea trainerilor.

Feedback din partea trainerilor

Super experienta, studentii au fost foarte curiosi si dornici de invatare.

Feedback din partea studentilor

Momentan nu s-a primit feedback din partea studentilor

Fig. 22 – Feedback general împărțit pe categorii

4.2.2 Detalii de implementare

Promovarea utilizatorilor la un anumit rol (admin sau trainer) se face în timp real datorită Firebase. Când acea componentă este inițializată, apelez o referință (*users*) creată de mine la baza de date cu toți utilizatorii (ca în Fig. 23 -), după care setez userii în store-ul de useri al aplicației creat cu MobX, astfel că orice modificare are loc pe lista de utilizatori se va afișa în pagină în timp real.

```
firebase.users().on('value', snapshot => {  
  
  // setez utilizatorii în store-ul MobX  
  
  setUsers(snapshot.val());  
  
  this.setState({ isLoading: false });  
  
});
```

Fig. 23 - Citire utilizatori în timp real din Firebase

După ce un utilizator a primit rolul de trainer, acesta va apărea în lista de traineri în momentul în care adminul va dori să asigneze un trainer unui anumit curs.

Concluzii și direcții viitoare

Această aplicație a avut ca scop principal extinderea funcționalității aplicației “FII Practic”, oferind utilizatorilor posibilitatea centralizării mai multor date precum și digitalizarea unor procese care se fac încă manual.

Având în vedere că proiectul “FII Practic” continuă să se dezvolte constant, și în aplicația T-Planner se pot dezvolta mai multe funcționalități de-a lungul timpului. În acest sens, consider că integrarea cu datele de pe serverul actual al aplicației “FII Practic” nu semnifică o problemă majoră și nu implică schimbarea logicii implementării prea mult.

Ca direcții viitoare propun posibilitatea generării rapoartelor în format PDF pentru liste cu informații care ar fi necesare sau ar fi de ajutor și în format fizic, nu doar în format digital, cum ar fi listele cu aplicanții ce urmează a fi contactați.

De asemenea, procesul de acordare a feedback-ului atât de la participanți cât și de la angajații companiilor de IT cu rol de trainer ar putea fi transformat într-un sistem cu acordare de note sau punctaj pe anumite topicuri relevante, iar în urma acestor punctaje să se obțină o medie a părerilor, o notă finală și o vizualizare de ansamblu a feedback-ului pentru ediția respectivă a proiectului.

O altă idee de dezvoltare ar fi implementarea unor grafice care să evidențieze diferențele statistice și progresul proiectului de la an la an, păstrând datele din edițiile anterioare. Consider că aceste informații ar fi de mare ajutor pentru următorii coordonatori ai proiectului în momentul în care vor planifica desfășurarea ediției următoare.

Personal, consider că aplicația “T-Planner” și-a atins obiectivul principal, acela de a folosi cât mai puține instrumente pentru comunicarea informațiilor între compania de IT, organizatorii proiectului și participanți și centralizarea mai multor date într-o singură aplicație. De asemenea, aceasta oferă posibilitatea de a se continua dezvoltarea, adaptarea și îmbunătățirea unor funcționalități precum și crearea altora noi.

Bibliografie

- Documentație React:
<https://reactjs.org/>
- Documentație Firebase:
<https://firebase.google.com/docs/web/setup/>
- Documentație MobX:
<https://mobx.js.org/>
- Documentație Formik:
<https://jaredpalmer.com/formik/docs/overview>
- Autentificare cu Firebase în React:
<https://www.robinwieruch.de/complete-firebase-authentication-react-tutorial/>
- MobX în React:
<https://www.robinwieruch.de/react-firebase-mobx-tutorial/>