

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Platformă pentru crearea prezentărilor
interactive: FrameIT**

propusă de

Cristian Lapteacru

Sesiunea: iunie/iulie, 2023

Coordonator științific

Lect. Dr. Vârlan Cosmin

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

**Platformă pentru crearea prezentărilor
interactive: FrameIT**

Cristian Lapteacru

Sesiunea: iunie/iulie, 2023

Coordonator științific

Lect. Dr. Vârlan Cosmin

Avizat,
Îndrumător lucrare de licență,
Lect. Dr. Vârlan Cosmin.

Data: Semnătura:

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Lapteacru Cristian** domiciliat în **Republica Moldova, mun. Chișinău, or. Durești, str-la Renașterii, nr. 6, bl. B, et. 3, ap. 12**, născut la data de **02 septembrie 2001**, identificat prin CNP **2006036045974**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2023, 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 **Platformă pentru crearea prezentărilor interactive: FrameIT** elaborată sub îndrumarea domnului **Lect. Dr. Vârlan Cosmin**, pe care urmează să o susțin 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 ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă 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:

Semnătura:

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Platformă pentru crearea prezentărilor interactive: FrameIT**, 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.

Absolvent **Cristian Lapteacru**

Data:

Semnătura:

Cuprins

Introducere	2
Motivație	2
Contribuție	2
Aspecte generale	2
Aspectul vizual	3
Securitatea	3
Testarea	3
Gradul de noutate	4
Structura lucrării	5
1 Tehnologii utilizate	6
1.1 JavaScript	6
1.2 ReactJS	7
1.3 Biblioteci externe	8
1.3.1 Quill	8
1.3.2 React-RND	9
1.4 MySQL	9
1.5 Concluzii	10
2 Fundamentele dezvoltării aplicației	11
2.1 Arhitectura aplicației	11
2.2 Arhitectura bazei de date	14
2.2.1 Aspecte generale	14
2.2.2 Structura bazei de date	14
2.3 Securitatea datelor	17
3 Utilizarea aplicației	18

4	Detalii de implementare	25
4.1	Componentele de autentificare și înregistrare	25
4.2	Componenta de administrare a contului	26
4.3	Componenta de creare și editare a proiectelor	28
4.3.1	Meniul	29
4.3.2	Bara de instrumente	31
4.3.3	Zona de editare	33
	Concluzii	35
	Bibliografie	36

Introducere

Motivație

Tema lucrării de licență reprezintă o platformă web simplificată pentru crearea prezentărilor interactive, numită FrameIT. Am dezvoltat o platformă simplificată pentru a face procesul de creare a prezentărilor mai accesibil și mai ușor pentru utilizatori.

FrameIT reprezintă o platforma intuitiva și ușor de înțeles. Utilizând interfețe ușor de utilizat și funcții generale, utilizatorii pot construi rapid și eficient prezentări interactive fără cunoștințe tehnice avansate.

Platforma a fost dezvoltată pentru a gestiona și a economisi timpul mai eficient. Iar pentru cei care nu au o anumită experiență la nivel tehnic să fie mult mai accesibilă.

Utilizatorii pot economisi timp și pot obține rezultate mult mai rapid utilizând procesul simplificat de creare a prezentărilor prin Platforma FrameIT. În loc să înceapă de la zero sau să folosească instrumente complicate, aceștia pot folosi funcționalitatea platformei și limitele predefinite pentru a crea rapid prezentări interactive de cel mai înalt nivel.

Principiile de accesibilitate și de utilizare ale platformei presupun facilitarea utilizării serviciilor pentru persoanele fără prea multe cunoștințe la nivel tehnic.

Contribuție

Aspecte generale

Contribuția mea la platformă este unică și reprezintă propriile valori și perspective în dezvoltarea platformei FrameIT. Printre aspectele principale care îmi fac contribuția diferită, dar în același timp valoroasă sunt:

- rezolvarea unor probleme importante: economisirea și gestionarea timpului;
- accesibilitatea pentru toți utilizatorii, cu sau fără experiență tehnică;

- eliminarea complexității tehnice.

Aspectul vizual

O altă contribuție este crearea unei interfețe atractive pentru utilizatori. Am folosit principii de design, bazate pe alegerea gamei coloristice, fonturilor și a structurii. Utilizarea unei game bine gândite de culori și elemente de design în întreaga platformă creează o experiență plăcută pentru utilizatori și ajută la transmiterea unui mesaj coerent prin intermediul platformei. Culorile au puterea de a evoca emoții și de a crea asociații în mintea utilizatorilor. Am optat pentru o gamă coloristică cu tentă neutră:

- Albastrul: sugerează liniște și profesionalism;
- Bej: sugerează energie și nu distrage atenția utilizatorului.

Culorile alese oferă un contrast adecvat pentru a facilita citirea și navigarea utilizatorilor pe platforma FrameIt.

Securitatea datelor

Încă o contribuție este securitatea datelor utilizatorilor și performanța. Am optat pentru implementarea măsurilor de securitate și protecție prin intermediul access token-urilor, întrucât securitatea datelor este îmbunătățită prin:

- autentificarea și autorizarea utilizatorilor;
- controlul accesului la resursele protejate;
- limitarea privilegiilor și protecția transferului de date.

Este important să se implementeze practici de securitate pentru generarea, gestionarea și utilizarea access token-urilor, pentru a asigura protecția datelor sensibile. Access token-ul joacă un rol crucial în asigurarea securității datelor într-o platformă. Prin intermediul unui access token, se stabilește și se validează identitatea utilizatorului și se controlează accesul la resursele protejate.

Testarea

Rezolvarea erorilor face parte din contribuțiile care duc la rezolvarea problemelor de utilizare a platformei în cadrul unei prezentări și aduc o contribuție valoroasă în crearea unei platforme stabile și de încredere. Atunci când erorile sunt identificate și

corectate, platforma devine mai stabilă, nu există riscul să se blocheze atât de des sau să prezinte diverse defecțiuni. Prin rezolvarea erorilor platforma:

- va îmbunătăți experiența utilizatorilor;
- nu vor exista nemulțumiri;
- va crește încrederea în platformă;
- va crește fidelitatea utilizatorilor pe termen lung.

Gradul de noutate

Tema platformei FrameIT pentru crearea prezentărilor interactive este una actuală și relevantă în contextul actual al tehnologiei și comunicării digitale.

Prezentările interactive sunt din ce în ce mai utilizate în majoritatea domeniilor și mediilor. Ele reprezintă o modalitate eficientă de a transmite informații complexe. Platforma FrameIT răspunde nevoii actuale de a oferi utilizatorilor o experiență interactivă față de prezentările tradiționale.

Odată ce tehnologia web a avansat, tot mai multe activități sunt realizate pe platforme web. Platforma FrameIT se adresează nevoii de a oferi o experiență fluidă și accesibilă printr-o interfață web intuitivă și ușor de utilizat în ceea ce privește crearea și editarea prezentărilor.

În contextul actual al tehnologiei există deja numeroase platforme pentru crearea prezentărilor. Cu toate acestea, concurența pe piață cere și o îmbunătățire continuă. Tehnologia web se dezvoltă rapid, aducând noi posibilități și tendințe în crearea de platforme.

Printre platformele populare și cunoscute de pe piață se numără Google Slides, Canva și PowerPoint Online. Aceste platforme oferă o gamă extinsă de instrumente complexe, care pentru persoane fără sau cu puțină experiență în domeniu pot părea complicate în utilizare. În consecință, interfețele acestor platforme pot deveni aglomerate și dificil de navigat pentru utilizatorii neexperimentați.

De aceea platforma FrameIT încearcă să ofere utilizatorilor o experiență plăcută și să răspundă la nevoile lor. Interfața simplificată a platformei date este obținută prin eliminarea complexității tehnice a platformelor curente de pe piață, fără a elimina instrumentele de bază necesare pentru crearea unei prezentări interactive. Astfel, pentru utilizarea platformei FrameIT nu sunt necesare careva cunoștințe avansate și permite oricui să creeze prezentări profesionale într-un mod eficient.

Structura lucrării

Această lucrare este structurată în 4 capitole principale, fiecare abordând aspecte specifice legate de aplicația web FrameIT.

- **Capitolul 1 "Tehnologii Utilizate"** : Acest capitol cuprinde o scurtă descriere a tehnologiilor utilizate pentru dezvoltarea aplicației FrameIT, dar și aspectele principale datorită cărora au fost alese aceste tehnologii.
- **Capitolul 2 "Fundamentele dezvoltării aplicației"** : Acest capitol abordează în detaliu arhitectura aplicației și a bazei de date folosite. În această parte se mai prezintă și instrumentele folosite pentru garantarea confidențialității datelor utilizatorilor.
- **Capitolul 3 "Utilizarea aplicației"** : Acest capitol descrie modul de utilizare și funcționalitățile aplicației FrameIT.
- **Capitolul 4 "Detalii de implementare"** : Acest capitol prezintă detaliile de implementare ale unor funcționalități și componente din aplicația FrameIT.
- **Concluzii** : În această parte a lucrării de licență se regăsesc cele mai importante idei din lucrare și potențiale direcții de viitor ale aplicației FrameIT.

Capitolul 1

Tehnologii utilizate

1.1 JavaScript

JavaScript este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor, utilizat în mod predominant în dezvoltarea aplicațiilor web. Printre aspectele principale care m-au facut să aleg acest limbaj de programare sunt:

- versatilitatea în dezvoltare;
- ecosistemul bogat;
- compatibilitatea cu toate browserele
- suportul comunității

Un motiv pentru care am optat pentru utilizarea acestui limbaj în dezvoltarea aplicației web, FrameIT este flexibilitatea și versatilitatea sa. JavaScript poate fi aplicat atât în dezvoltarea părții de vizualizare a aplicației (front-end), cât și în cea de gestionare a datelor și logicii (back-end), folosind mediul de dezvoltare Node.js. Acest lucru facilitează transferul eficient al datelor și a logicii între componentele care se ocupă de interfața cu utilizatorul și cele care gestionează procesarea și stocarea acestor date.

Încă un argument pentru alegerea javascript ar fi ecosistemul său bogat. Acest limbaj dispune de o gamă largă de biblioteci și instrumente publice care ajută la accelerarea procesului de dezvoltare. Pentru activități tipice, cum ar fi gestionarea modificării DOM, gestionarea stării, efectuarea de interogări API și gestionarea interacțiunilor utilizatorilor, aceste biblioteci oferă soluții gata făcute. O bibliotecă bine cunoscută este React.js, ea oferă o mulțime de funcționalități și suport comunității. Această bibliotecă a fost folosită pentru realizarea interfeței grafice a platformei.

Astăzi javascript este utilizat pe scară largă ca limbaj standard pentru dezvoltarea

aplicațiilor web. Are o comunitate de dezvoltatori considerabilă și vibrantă și este acceptată de toate browserele web principale. Aceasta garantează compatibilitatea între platforme și suport pentru aplicațiile online.

Dintre limbajele de programare, JavaScript are una dintre cele mai mari și mai vibrante comunități. Comunitatea JavaScript este foarte utilă și cooperantă, oferind o multitudine de instrumente, documentații, forumuri și biblioteci pentru a ajuta dezvoltatorii.

1.2 ReactJS

ReactJS este o bibliotecă JavaScript publică creată de Facebook care este utilizată în dezvoltarea interfețelor grafice pentru aplicațiile web. Printre aspectele principale care m-au făcut să aleg această librărie pentru crearea interfeței grafice a platformei FrameIT sunt:

- eficiența și performanța;
- reutilizarea componentelor;
- ecosistemul bogat;

ReactJS folosește un DOM virtual, care este o replică compactă a DOM-ului real din memorie, pentru a permite randarea eficientă. Această metodă prezintă o serie de avantaje.

Principalul avantaj este că reduce cantitatea de manipulări directe și costisitoare la DOM prin actualizarea eficientă doar a zonelor care sunt necesare, îmbunătățind performanța și experiența utilizatorului.

În al doilea rând, React elimină dificultățile de a manipula direct DOM-ul, care poate fi predispus la erori și dificil de întreținut, prin utilizarea unui DOM virtual. React se ocupă de actualizări și rezolvă rapid orice discrepanțe între DOM-urile virtuale și reale, menținând interfața de utilizare sincronizată.

Prin arhitectura sa bazată pe componente, ReactJS încurajează reutilizarea. O anumită parte a interfeței și funcționalitatea aferentă acestora sunt conținute în unități de construcție autonome cunoscute sub numele de componente. Aceste piese pot fi utilizate în diferite zone ale programului sau chiar în mai multe proiecte. Prin încurajarea modulară este economisit timpul și efortul dezvoltatorilor. În plus, prin reutilizare se

îmbunătățește întreținerea. La necesitate, dezvoltătorii nu trebuie să modifice toată aplicația, dar doar componenta dorită. Ulterior, componenta actualizată este utilizată automat peste tot. Acest lucru scade posibilitatea de a introduce erori și face gestionarea și actualizarea programului în ansamblu mai simplă.

ReactJS dispune de un ecosistem puternic și bogat care este format dintr-o varietate de biblioteci, instrumente și suport comunitar. Aceste biblioteci îmbunătățesc procesul de dezvoltare, prin oferirea soluțiilor la unele probleme deja existente. Astfel, este economisit timp și efort pentru dezvoltarea aplicațiilor sofisticate. Pentru dezvoltarea aplicației FrameIT au fost folosite mai multe biblioteci externe populare precum React Router pentru gestionarea navigației, Axios pentru gestionarea apelurilor API, React-Quill și React-RND pentru gestionarea componentei de creare și editare a prezentărilor.

1.3 Biblioteci externe

1.3.1 Quill

Quill este o bibliotecă JavaScript care oferă un editor de text bogat, robust și flexibil pentru aplicații web. Similar cu editoarele de text binecunoscute precum Microsoft Word sau Google Docs, permite utilizatorilor să genereze, să formateze și să editeze conținutul dorit. O varietate de funcții sunt disponibile în editorul Quill, cum ar fi liste, anteturi, opțiuni de stil de text și multe altele. Pentru o interacțiune ușoară cu aplicațiile ReactJS, a fost creat React-Quill. Acesta oferă o componentă de editor Quill pentru React, facilitând includerea editorului și controlul stării acestuia în componentele ReactJS.

Aspectul principal care m-a făcut să aleg această librărie pentru componenta de creare și editare a prezentărilor, mai exact, a textului din slide-uri a acestei componente este nivelul ridicat de personalizare și extensibilitate a React-Quill. De exemplu, pentru a vedea în timp real actualizarea textului din slide, am folosit o bară de instrumente proprie care comunică cu componenta React-Quill și îi transmite informațiile necesare.

Un alt motiv pentru alegerea acestei biblioteci ar fi compatibilitatea și suportul web, care nu sunt oferite de toate librării identice. Editorul de bază al React-Quill, Quill, a fost supus unor teste și optimizări semnificative pentru a fi compatibil cu o

gamă largă de browsere, oferind o experiență consistentă pe multe platforme.

1.3.2 React-RND

React-Rnd este o binecunoscută librărie ReactJS care oferă componente care pot fi glisate și redimensionabile, permițând utilizatorilor să redimensioneze și să mute în mod interactiv obiectele de pe ecran. Oferă un API simplu și adaptabil pentru adăugarea de funcționalități de tragere și redimensionare la aplicațiile ReactJS. Utilizatorii pot prinde marginile sau colțurile acestor componente pentru a le redimensiona și pot face click și trage pentru a le muta pe ecran.

Aspectul principal care m-a făcut să aleg această librărie pentru componenta de creare și editare a prezentărilor, mai exact, a imaginilor și textului din slide-uri este faptul că ea este unica care permite atât glisarea cât și redimensionarea obiectelor într-o aplicație ReactJS, pe când restul bibliotecilor oferă doar câte o opțiune și nu permit adăugarea alteia.

1.4 MySQL

Pentru menținerea și organizarea datelor structurate am folosit MySQL, un sistem public de gestionare a bazelor de date relaționale. Unul dintre cele mai utilizate sisteme de baze de date la nivel mondial, este renumit pentru fiabilitatea, eficiența și scalabilitatea sa. Printre aspectele principale care m-au făcut să aleg acest sistem sunt:

- ușurința în utilizare și administrare;
- performanță, stabilitate și viteză;

Sintaxă oferită de sistemul MySQL este simplă și intuitivă, ușor de înțeles și de utilizat. Acesta oferă, de asemenea, instrumente de administrare puternice, cum ar fi phpMyAdmin și MySQL Workbench, care facilitează administrarea bazei de date și interacțiunea cu aceasta.

MySQL este utilizat în producție de mulți ani, ceea ce l-a făcut un sistem stabil și de încredere. Prezentările utilizatorilor aplicației FrameIT sunt stocate în baza de date cu ușurința datorită faptului că MySQL a fost proiectat pentru a gestiona volume mari de date. O serie de factori contribuie la performanța și rapiditatea excepționale ale

MySQL. Eficiența prelucrării și a extragerii datelor este sporită prin indexarea eficientă și optimizarea interogărilor. Datele care sunt adesea preluate sunt păstrate în memorie datorită mecanismele de stocare cache, care accelerează execuția interogărilor. Astfel, aplicațiile primesc un răspuns la cereri într-un timp eficient.

1.5 Concluzii

Tehnologiile utilizate pentru dezvoltarea aplicației web FrameIT, au fost alese atât pentru a satisface cerințele aplicației FrameIT, cât și pentru a-mi spori dezvoltarea personală în domeniul dezvoltării web.

Prin alegerea JavaScript ca limbaj de programare pe partea de backend și a mediului de dezvoltare Node.js, am putut construi logica de server și gestiona interacțiunea cu baza de date și alte componente ale aplicației. JavaScript oferă o sintaxă expresivă și puternică, facilitând dezvoltarea serverelor și a API-urilor web. Utilizând JavaScript atât pe backend, cât și pe frontend, am putut lucra cu aceeași limbaj de programare, simplificând dezvoltarea și integrarea componentelor aplicației.

Pe partea de frontend, am ales ReactJS pentru a construi interfața utilizator a aplicației FrameIT. ReactJS este o bibliotecă JavaScript populară și puternică, care mi-a permis să obținut un cod mai clar și ușor de întreținut, facilitând dezvoltarea și scalabilitatea aplicației. Datorită ecosistemului bogat am putut beneficia de bibliotecile externe care au facilitat dezvoltarea aplicației.

Alegerea de a utiliza JavaScript și ReactJS în dezvoltarea aplicației FrameIT mi-a oferit oportunitatea de a crește și de a evolua ca programator. Pe măsură ce am lucrat cu aceste tehnologii în timpul proiectului, am învățat multe lucruri noi și am dobândit abilități importante în dezvoltarea cu JavaScript. De asemenea, am avut ocazia să descopăr și să mă familiarizez cu biblioteca ReactJS.

În concluzie, tehnologiile JavaScript și ReactJS utilizate în dezvoltarea aplicației FrameIT au fost alese pentru a îndeplini cerințele funcționale ale aplicației, în timp ce oferă oportunități de dezvoltare și creștere personală. Am dobândit experiență în dezvoltarea cu aceste tehnologii, pregătindu-mă pentru a aborda și alte proiecte web complexe în viitor.

Capitolul 2

Fundamentele dezvoltării aplicației

2.1 Arhitectura aplicației

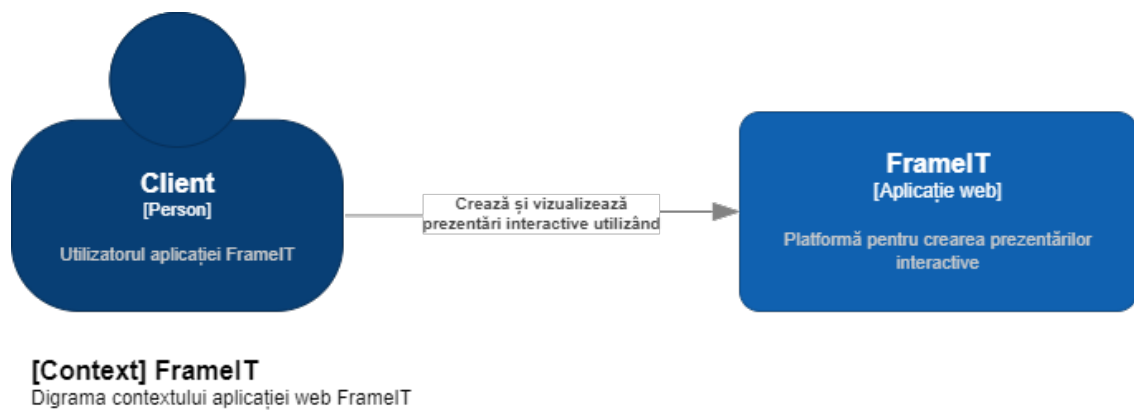


Figura 2.1: Diagrama C4 nivelul 1 al aplicației FrameIt

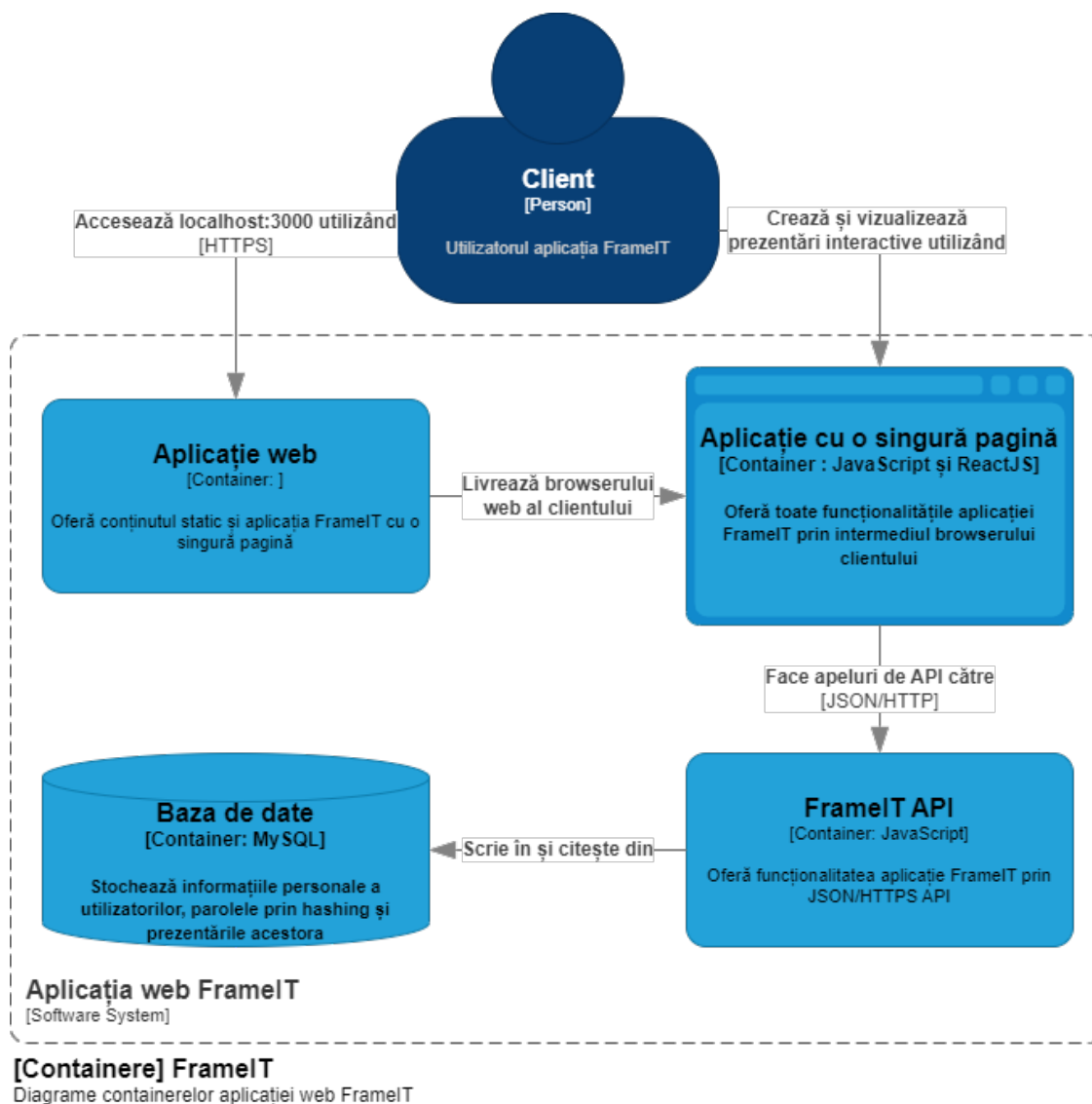


Figura 2.2: Diagrama C4 nivelul 2 al aplicației FrameIT

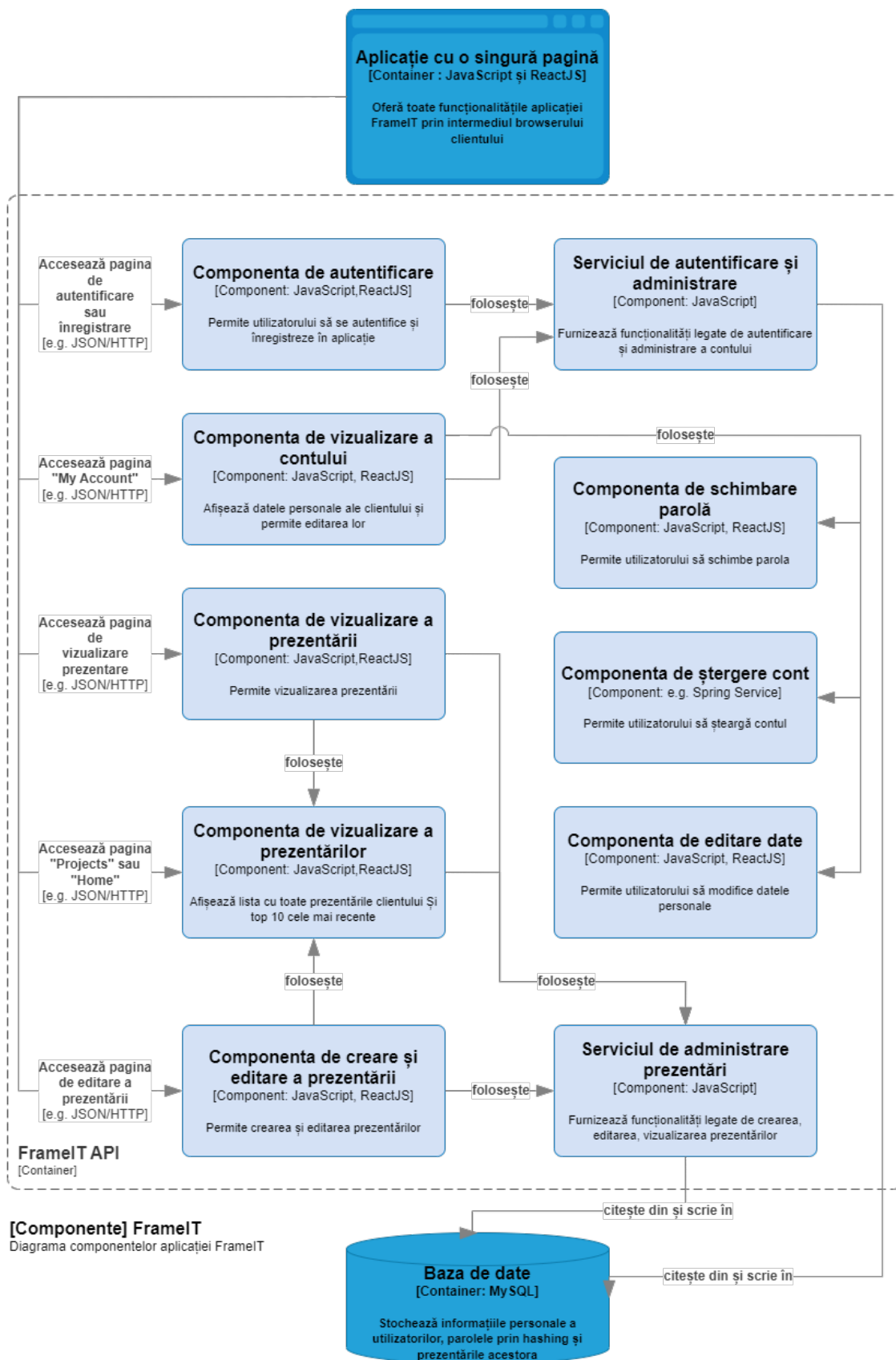


Figura 2.3: Diagrama C4 nivelul 3 al aplicației FrameIt

2.2 Arhitectura bazei de date

2.2.1 Aspecte generale

Pentru stocarea și gestionarea datelor necesare funcționalităților aplicației, FrameIT utilizează o bază de date relațională, MySQL. Conexiunea aplicației cu baza de date MySQL a fost efectuată cu ajutorul librăriei javascript "mysql", mai exact cu ajutorul funcției "createConnection". Ulterior interogările au fost efectuate cu ajutorul funcției "query".

```
1 import mySql from "mysql"
2 //Crearea conexiunii cu baza de date
3 const db=mySql.createConnection({
4   host:"localhost",
5   user:"root",
6   password:"*****",
7   database:"FrameIT"
8 })
```

Figura 2.4: Conexiunea cu baza de date MySQL

2.2.2 Structura bazei de date

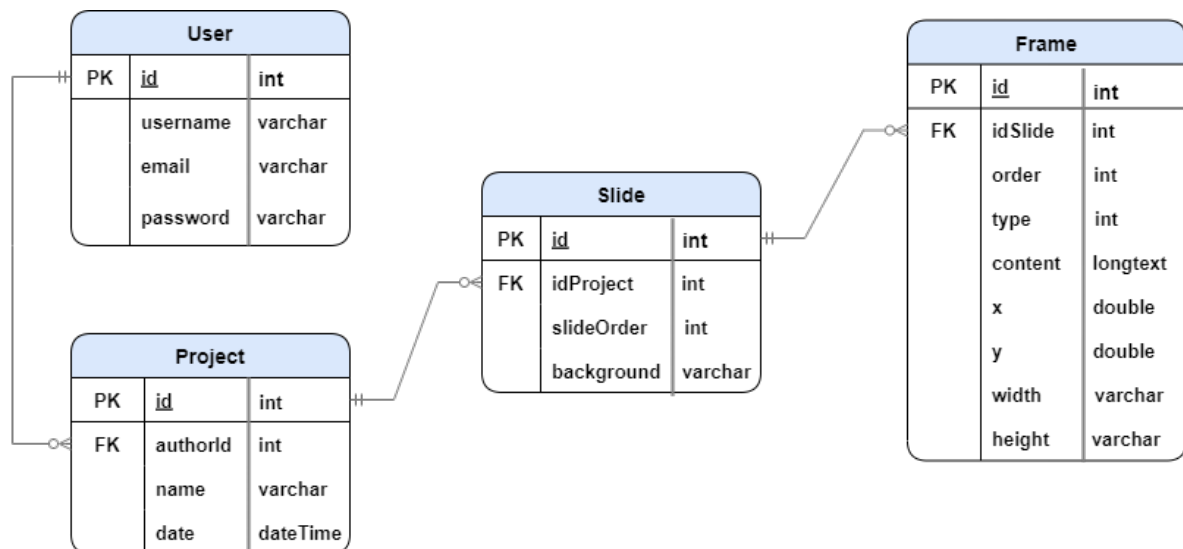


Figura 2.5: Schema bazei de date utilizate în aplicația FrameIT

Conform figurei 2.1 baza de date a aplicației este formată din 4 tabele relaționate

între ele prin intermediul cheilor primare și celor externe. fiecare având un rol important în funcționarea corectă a aplicației. Pentru a oferi posibilitatea unui utilizator de a crea prezentări interactive ce conțin numeroase slide-uri și containere text și imagini au fost folosite următoarele relații de unu-la-mai-muți:

- relația între tabelul "Users" și tabelul "Projects", Această relație permite unui utilizator să creeze mai multe prezentări interactive. Coloana cheie primară "id" din tabela "Users" este utilizată ca și cheie străină "authorId" în tabela "Projects" pentru a identifica prezentările asociate fiecărui utilizator.
- relația între tabelul "Projects" și tabelul "Slides", Această relație permite adăugarea mai multor slide-uri într-o prezentare. Coloana cheie primară "id" din tabela "Projects" este utilizată ca și cheie străină "idProject" în tabela "Projects" pentru a identifica slide-urile asociate fiecărei prezentări.
- relația între tabelul "Slides" și tabelul "Frames", Această relație permite adăugarea mai multor containere text și imagini în cadrul fiecărui slide. Coloana cheie primară "id" din tabela "Slides" este utilizată ca și cheie străină "idSlide" în tabela "Frames" pentru a identifica containerele asociate fiecărui slide.

Tabelul "Users" stochează informațiile personale a utilizatorilor. El este necesar atât pentru partea de autentificare, cât și pentru partea de stocare a prezentărilor, deoarece aceasta este unica metoda de a identifica autorul prezentării. Tabelul este format din 4 coloane:

- id: aceasta este coloana cheie primară a tabelului, care identifică în mod unic fiecare înregistrare a utilizatorului.
- username: această coloană stochează numele utilizatorului, care poate fi utilizat în procesul de autentificare pentru a identifica și autentifica utilizatorul în aplicație.
- email: această coloană stochează emailul utilizatorului, care poate fi utilizat în procesul de autentificare pentru a identifica și autentifica utilizatorul în aplicație.
- password: această coloană stochează parola utilizatorului, care este criptată pentru a asigura securitatea datelor utilizatorilor. Parola este utilizată în procesul de autentificare pentru a verifica identitatea utilizatorului.

Tabelul "Projects" stochează datele aferente prezentărilor utilizatorilor și este format din 4 coloane:

- id: aceasta este coloana cheie primară a tabelului. Fiecare prezentare are un identificator unic atribuit automat. Acest identificator servește la identificarea fiecărei prezentări în cadrul tabelului.
- authorId: aceasta este o cheie secundară care face referință la coloana "id" din tabelul "Users". Relația între aceste două tabele permite identificarea autorului fiecărei prezentări.
- name: această coloană stochează numele prezentării, pentru a o identifica și diferenția de celelalte prezentări.
- date, această coloană conține data și ora când prezentarea a fost creată sau editată ultima oară, pentru a diferenția prezentările recente.

Tabelul "Slides" stochează datele legate de slide-urile prezentărilor și este format din 4 coloane:

- id: aceasta este coloana cheie primară a tabelului. Fiecare slide are un identificator unic atribuit automat. Acest identificator servește la identificarea fiecărui slide în cadrul tabelului.
- idProject: aceasta este o cheie secundară care face referință la coloana "id" din tabelul "Projects". Relația între aceste două tabele permite identificarea proiectului căruia îi aparține fiecare slide.
- slideOrder: această coloană stochează ordinea slide-ului în prezentare.
- background: această coloană stochează culoarea slide-ului.

Tabelul "Frames" este format din 9 coloane în care se stochează datele aferente containerelor de text și imagini care aparțin slide-urilor. Aceste date sunt necesare pentru reconstruirea prezentărilor create de utilizatori pentru a le oferi posibilitatea de a edita sau vizualiza prezentările. Coloanele acestui tabel sunt:

- id: aceasta este coloana cheie primară a tabelului. Fiecare container, numit și "frame" are un identificator unic atribuit automat. Acest identificator servește la identificarea fiecărei container în cadrul tabelului.

- `slideId`: aceasta este o coloana cheie secundară care face referință la coloana "id" din tabelul "Slides". Relația între aceste două tabele permite identificarea slide-ului căruia îi aparține fiecare container.
- `order`: această coloană păstrează ordinea containerelor de tip text sau imagine în cadrul slide-ului.
- `type`: această coloană reține un index pentru tipul containerului. Există două tipuri de containere: container text care are indexul 1 și container imagine care are indexul 2.
- `content`, aceasta coloană poate include textul containerului text în format de cod html pentru a păstra stilurile lui sau reprezentarea imaginii convertite la un șir în baza 64.
- `x`: reține poziția containerului pe orizontală, axa X în cadrul slide-ului.
- `y`: reține poziția containerului pe verticală, axa Y în cadrul slide-ului.
- `width`: această coloană păstrează lungimea containerului și este stocată în format varchar, de exemplu, "100px"..
- `height`: această coloană păstrează înălțimea containerului și este stocată în format varchar, de exemplu, "100px".

2.3 Securitatea datelor

Capitolul 3

Utilizarea aplicației

Modul în care interacționează utilizatorul cu aplicația este un aspect crucial pentru platforma FrameIT. Navigarea și interacțiunea utilizatorului au fost gândite să fie intuitive, ușor de înțeles și eficiente. Consider că o experiență utilizator fluidă și plăcută poate contribui în mod semnificativ la succesul unei aplicații.

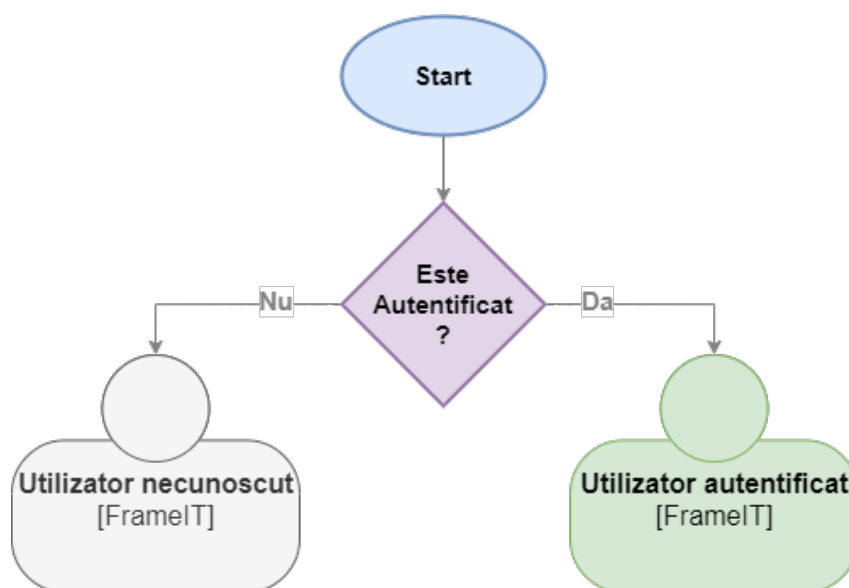


Figura 3.1: Diagrama interacțiunii utilizatorului cu aplicația FrameIT

La accesarea platformei FrameIT se verifică dacă utilizatorul este autentificat sau nu. În caz afirmativ, el este redirecționat către pagina "Home" cu funcționalitățile depline ale aplicației. Dacă utilizatorul nu este autentificat va fi redirecționat la pagina "Home" cu funcționalități limitate.

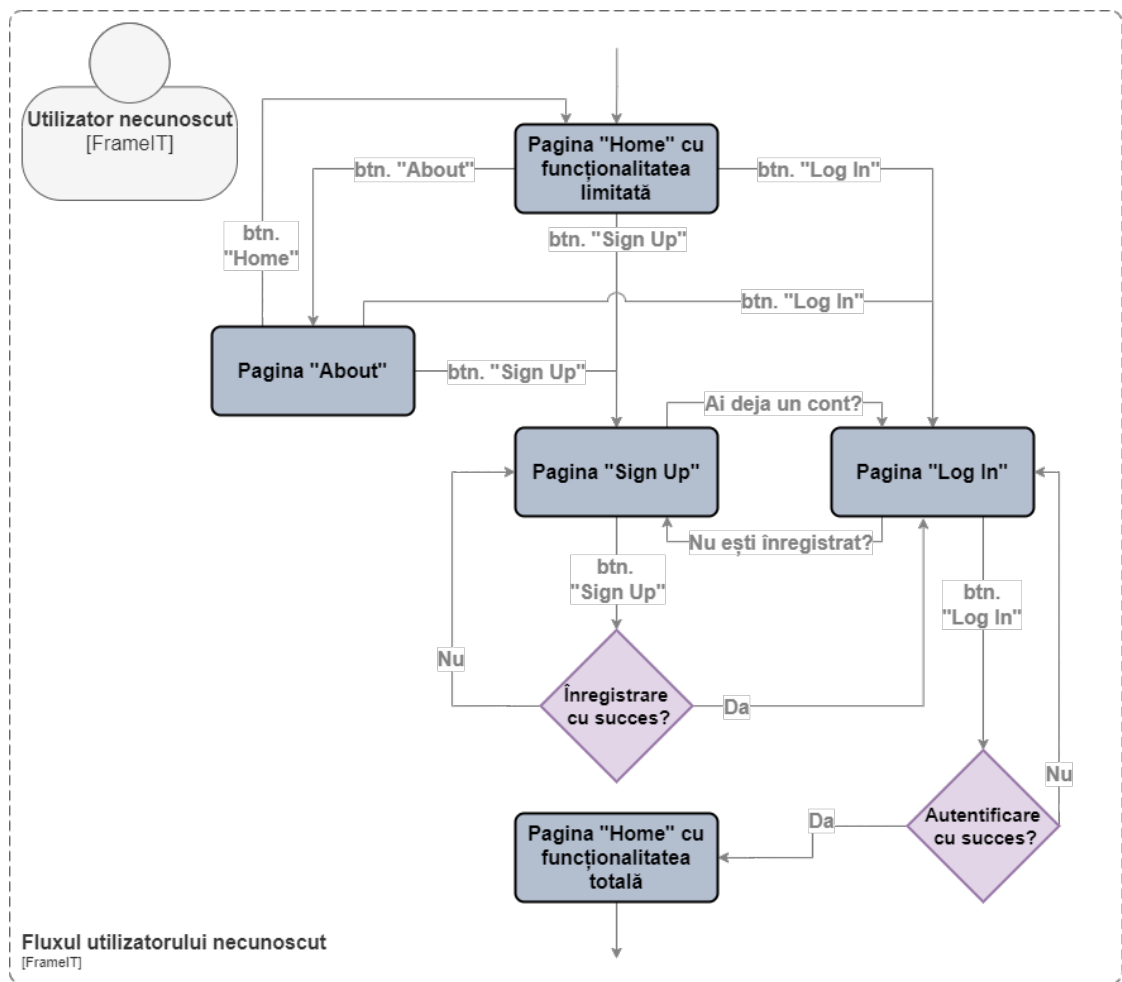


Figura 3.2: Diagrama interacțiunii utilizatorului necunoscut cu aplicația

În cazul în care utilizatorul nu este autentificat el are acces doar la unele funcționalități ale aplicației. În situația dată el are dreptul doar să se informeze despre aplicația FrameIT și să se înregistreze sau să se autentifice. Utilizatorul poate naviga prin paginile:

- **"Home"** : Aici clientul poate găsi o descriere cu caracteristicile principale ale aplicației. Din pagina "Home" poate ajunge în paginile: "About", "Log In" cu ajutorul barei de navigare și în pagina "Sign Up" atât prin intermediul barei de navigare cât și prin accesarea butonului "Sign Up" de pe pagina "Home".
- **"About"** : În această pagină utilizatorul la fel se poate documenta despre aplicația FrameIT și funcționalitățile sale. Din pagina "About" cu ajutorul barei de navigare poate ajunge în paginile:
 - "Home"
 - "Log In"
 - "Sign Up"

- **"Sign Up"** : În această pagină persoana își poate crea un cont în aplicația FramelIT completând formularul care conține următoarele date: nume, email, parolă, confirmare parolă. Și în cazul în care datele au fost introduse corect și nu au fost careva probleme la verificarea datelor pe partea de backend de exemplu: parola este diferită de parola confirmată, numele sau emailul este deja folosit de un alt utilizator, utilizatorul este redirecționat către pagina "Log In". Altfel, utilizatorului îi sunt afișate erorile. Însă dacă persoană deja are un cont și nu dorește să își mai creeze încă unul, prin apăsarea pe textul: "Already registered?", persoana la fel este redirecționată către pagina "Log In", fără să completeze formularul de înregistrare.
- **"Log In"** : Aflându-se pe această pagină utilizatorul poate reveni la pagina "Sign Up" în cazul în care nu este înregistrat deja în aplicație prin intermediul apăsării textului "Not Registered yet?". În cazul în care utilizatorul este deja înregistrat îi rămâne doar să completeze formularul de autentificare ce conține următoarele câmpuri: nume și parolă. În cazul în care formularul este completat corect persoana din utilizator necunoscut va deveni utilizator autentificat și va fi redirecționat către pagina "Home" cu funcționalitățile depline, altfel utilizatorului îi va fi afișat un mesaj de eroare în dependență de problemă, de exemplu: la introducerea unei combinații dintre nume și parolă greșite, mesajul de eroare va fi: "This user does not exists!".

În cazul în care utilizatorul este autentificat, el are acces la toate funcționalitățile aplicației și poate naviga prin următoarele pagini:

- **"Home"** : Pe pagina principală utilizatorul autentificat pe lângă funcționalitățile disponibile unui utilizator necunoscut mai are dreptul la acces la vizualizarea prezentărilor sale recente, unde apăsând pe unul din proiectele lui poate edita, vizualiza și șterge proiectul. Mai mult, acum utilizatorul mai poate și crea noi proiecte apăsând pe butonul "New Project" care îl redirecționează către pagina "Edit Project". Astfel, acum utilizatorul din pagina curentă poate accesa următoarele pagini:
 - "About"
 - "Projects"

- "My Account"
 - "Edit Project"
 - "View Project".
- **"About"** : Această pagină nu oferă funcționalități noi, dar datorita faptului că bara de navigare se schimbă la autentificare, persoana poate accesa paginile:
 - "Home"
 - "Projects"
 - "My Account"
 - **"Projects"** : În pagina dată se afișează o listă cu toate proiectele utilizatorului, de unde la fel le poate edita, vizualiza și șterge. Din această pagină utilizatorul poate ajunge în paginile :
 - "Home"
 - "About"
 - "Edit Project"
 - "View Project"
 - "My Account".
 - **"My Account"** : Această pagină oferă funcționalități de vizualizare a informației personale și de editare a ei. Această pagină pe lângă meniul aplicației, are și meniul propriu, în care se găsesc următoarele butoane:
 - "Information"
 - "Change Password"
 - "Delete Account"
 - "Log Out"

La deschiderea paginii utilizatorul găsește informația personală pe care o și poate edita. În cazul în care dorește să modifice numele sau emailul, utilizatorul face modificările și apasă butonul "Save Changes". În backend se fac verificările și validările necesare, se verifică dacă chiar utilizatorul dat este conectat și cere modificările date și în cazul în care nu sunt găsite careva probleme pe ecranul utilizatorului îi va apărea un mesaj de validare, altfel, un mesaj de eroare.

La apăsarea butonului "Change Password" utilizatorului îi apare funcționalitatea de a schimba parola curentă. Pentru asta este necesar să completeze formularul cu următoarele câmpuri: parola curentă, parola nouă, confirma parola. Prin apăsarea butonului "Change" se verifică informația introdusă și în cazul în care nu se găsesc erori, de exemplu parola curentă greșită, utilizatorului îi este afișat un mesaj de validare, altfel un mesaj de eroare.

Butonul "Delete Account" oferă posibilitatea de a șterge contul de pe platforma FrameIT print introducerea parolei curente și apăsarea butonului "Delete". În cazul în care parola este corectă și dacă corespunde informațiile despre utilizator cu cele din tokenul de acces se transmite un mesaj de validare și persoana este redirecționată pe pagina "Home" cu funcționalitatea redusă, altfel un mesaj de eroare.

Butonul "Log out" permite deconectarea din aplicația FrameIT, la apăsare din utilizator autentificat persoana devine utilizator necunoscut și la fel este redirecționat către pagina "Home" cu funcționalitatea redusă.

Din această pagină puteți visita următoarele pagini:

- "Home"
- "About"
- "Projects"
- "MyAccount"
- **"Edit Project"**: Aceasta este pagina principală a platformei FrameIT, ea încorporează funcționalitățile de bază ale acestei platforme. Aici persoana poate crea și edita prezentări interactive. Utilizatorul are accesul la următoarele funcționalități:
 - Butonul "New Slide" : Adaugă un slide nou care devine slide-ul curent de lucru.
 - Butonul "Text" : Adaugă un nou container text în slide-ul curent.
 - Butonul "Image" : Adaugă o nouă imagine în slide-ul curent.
 - Butonul "Delete Frame" : Șterge containerul curentă din slide-ul curent.
 - Butonul "Edit slide" : Setează o culoare pentru fundalul slide-ului curent.
 - Instrumente de editare text : Editează textul din containerul text curentă, poate seta mărimea textului, culoarea, stilul și aliniera lui.

- Câmpul "Project Name" : Setează numele proiectului curent.
- Lista de slide-uri : Reprezintă o previzualizare a tuturor slide-urilor din proiectul curent și prin apăsarea pe unul din ele se modifica slide-ul curent, fapt evidențiat prin marginea de culoare albastră a previzualizării slide-ului curent. Mai mult slide curent poate fi mutat în sus și în jos pentru a schimba ordinea slide-urilor în proiect.
- La apăsarea containerelor din cadrul slide-ului se schimbă containerul curent "frame", fapt evidențiat prin marginea de culoare albastră a frame-ului curent. Aceste containere pot fi redimensionate prin apăsarea cu ajutorul cursorului pe colțul drept-jos sau latura lateral din partea dreaptă și tragerea. Mai mult, ele pot fi și mutate în interiorul slide-ului prin apăsarea containerului și tragerea lui.
- Butonul "Save" : salvează proiectul curent, în cazul în care nu sunt erori, se va afișa mesajul "Project saved!", altfel se va afișa mesajul de eroare: "Project Name already used!".
- **"View Project"**: În această pagină se vizualizează/prezintă proiectele. Slide-urile se schimbă cu ajutorul tastaturii. Pentru a ajunge la slide-ul anterior se folosește butonul cu săgeată stânga, iar pentru a ajunge la slide-ul următor se folosește butonul cu săgeată dreapta.

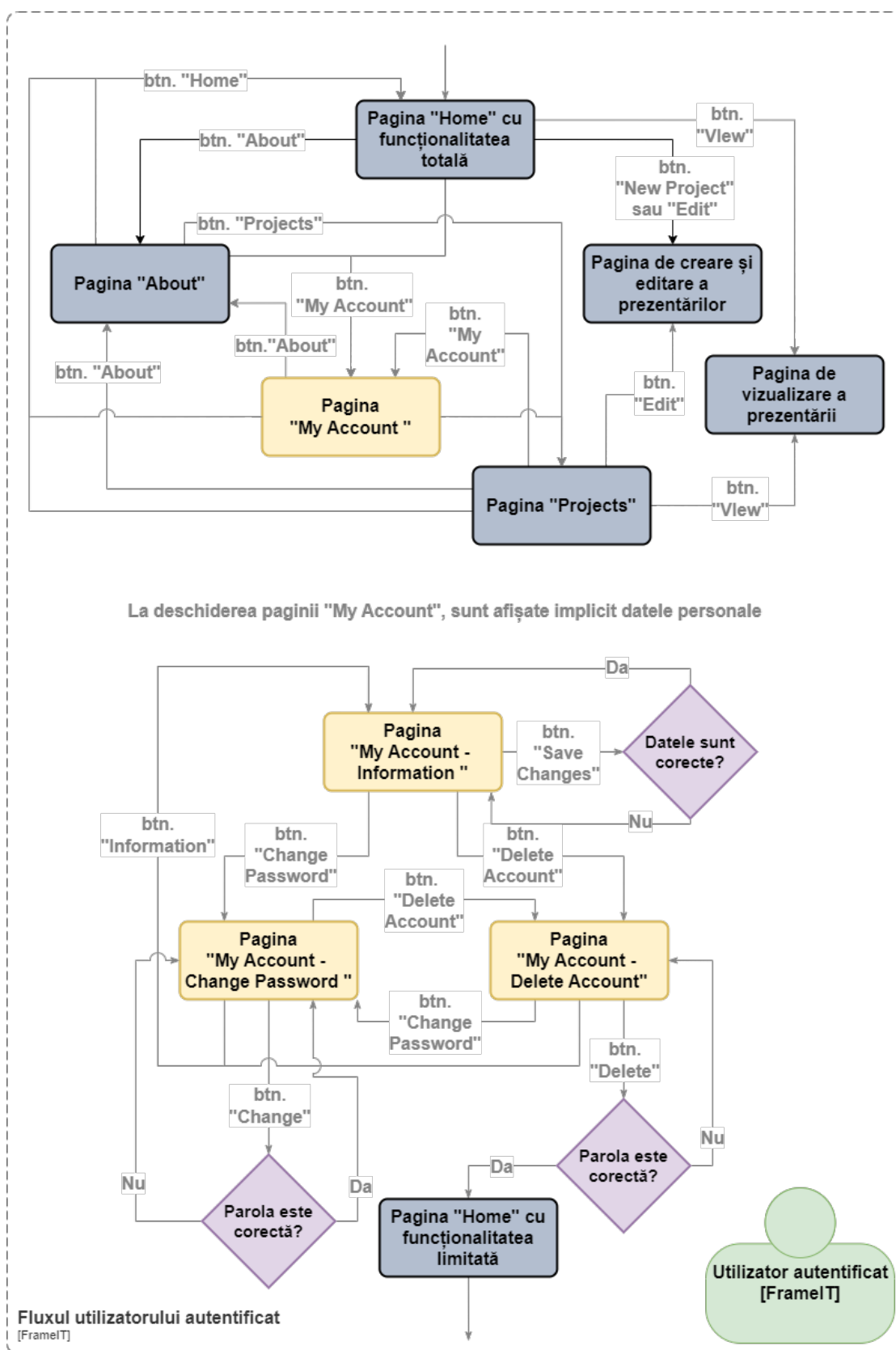


Figura 3.3: Diagrama interacțiunii utilizatorului autentificat cu aplicația FrameIT

Capitolul 4

Detalii de implementare

4.1 Componentele de autentificare și înregistrare

În aplicația FrameIT autentificarea și înregistrarea se realizează prin intermediul completării unui formular. Pentru fiecare câmp din formular există o variabilă care stochează conținutul lui. La fiecare modificare de conținut al unui câmp se actualizează valoarea variabilei respective. Formularul de înregistrare permite submiterea lui doar dacă câmpurile parolă și confirmare parolă sunt identice, altfel se afișează un mesaj de eroare. Ulterior datele sunt stocate într-un dicționar cu următoarele perechi de chei:

- "email", "password" : pentru formularul de autentificare.
- "username", "email", "password" : pentru formularul de înregistrare.

Apoi prin intermediul apelurilor de API :

```
1 //pentru formularul de autentificare
2 axios.post("/auth/login", inputs, {withCredentials:true, });
3 //pentru formularul de nregistrare
4 axios.post("/auth/signup", inputs)
```

se transmit datele serverului aplicației pentru verificarea informației.

În cazul înregistrării unui utilizator nou, serverul verifică disponibilitatea numelui și emailul, în cazul în care numele și emailul sunt deja inserate în baza de date, serverul trimite înapoi clientului un mesaj de eroare, altfel se criptează parola printr-o metodă unidirecțională și se stochează datele noului utilizator. Apoi, este transmis un mesaj de validare și clientul este redirecționat la pagina de autentificare.

În cazul autentificării unui utilizator, serverul verifică dacă există un utilizator în baza de date cu acest email. În caz că nu, serverul transmite înapoi un mesaj de eroare, altfel, se verifică corectitudinea parolei. Parola transmisă este criptată și comparată cu

parola criptată din baza de date. În cazul în care există diferențe, se transmite înapoi un mesaj de eroare, altfel se crează un token de acces folosind librăria "jsonwebtoken", în care se stochează id-ul utilizatorului. Ulterior se stochează acest token în cookies și se transmite un mesaj de validare, ce conține datele nevulnerabile ale utilizatorului.

```
1 import jwt from "jsonwebtoken"
2 router.post("/login", (req, res) => {
3     var q = "SELECT * FROM users WHERE email = ? "
4     db.query(q, [req.body.email], (err, result) => {
5         if (err)
6             return res.json(err)
7         if (result.length === 0)
8             return res.status(201).json("This user does not exist.")
9         if (!bcrypt.compareSync(req.body.password, result[0].password))
10            return res.status(201).json("Wrong Password.")
11         //Mai jos se creaza Access Tokenul si se stocheaza in Cookies
12         const token = jwt.sign({id: result[0].id}, "PASSWORD")
13         res.cookie("accessToken", token, {httpOnly: true,}).status(200).json({
14             id: result[0].id, username: result[0].username, email: result[0].email})
15     })
16 })
```

Figura 4.1: Crearea tokenului de acces.

4.2 Componenta de administrare a contului

Pagina de administrare a contului este formată din 3 componente:

- informații personale
- schimbare parolă
- ștergere cont

Editarea informațiilor personale, schimbarea parolă și ștergerea contului se realizează prin intermediul unui formular. Fiecărui câmp din formular îi este atribuită câte o variabilă care se actualizează la fiecare modificare. La început ele au valoare null. Formularul de schimbare parolă permite submiterea sa doar dacă valorile câmpurilor: parolă nouă și confirmă parola sunt identice, altfel va fi afișat un mesaj de eroare. Ulterior, datele sunt stocate într-un dicționar cu următoarele chei:

- "username", "email", "password", "userId" : pentru formularul de editare date personale.

- "oldPassword", "newPassword", "userId" : pentru formularul de schimbare parola.

- "password", "userId" : pentru formularul de ștergere cont

Apoi prin intermediul apelurilor de API :

```
1 //pentru formularul de editarea datelor personale
2 axios.post("/change/info", inputs, {withCredentials:true, });
3 //pentru formularul de schimbare parola
4 axios.post("/change/passw", inputs, {withCredentials:true, })
5 //pentru formularul de ștergere a contului
6 axios.post("/change/delete", inputs, {withCredentials:true, })
```

se transmit informațiile serverului.

Serverul decriptează tokenul de acces pentru a verifica identitatea utilizatorului, se compară id-ul utilizatorului primit de la client cu cel salvat în tokenul de acces. Dacă

```
1 //Functia actualizeaza informatia personala
2 router.post("change/info", (req, res)=>{
3     //Obtinerea Access Tokenului din Cookies
4     const accessToken = req.cookies.accessToken;
5     if (accessToken)
6         try {
7             //Decodificarea Access Tokenului
8             const decodedToken = jwt.verify(accessToken, '*****');
9             //Verificarea daca id trimis de client este indentic
10            //cu cel de la autentificare
11            if(req.body.userId === decodedToken.id){
12                //efectuarea verificarilor si a modificarilor in baza
13                //de date in caz de necesitate
14            }
15        }
16    })
```

Figura 4.2: Utilizarea tokenului de acces.

identitatea este confirmată, prin intermediul interogărilor se efectuează verificarea parolei. Parola primită de la client se criptează și se compară cu cea din baza de date. În cazul în care coincid, se fac actualizările necesare și se transmite un mesaj de validare,

altfel se transmite un mesaj de eroare.

4.3 Componenta de creare și editare a proiectelor

Pagina de creare și editare a proiectelor este pagina cu funcționalitățile principale ale aplicației FrameIT. Structura de date folosită pentru stocarea informației proiectului în partea de front-end este maparea imbricată (nested map). Această structură de date are următoarele chei:

- `id` : reprezintă ordinea slide-ului în proiect.
- `background` : reprezintă culoarea fonului slide-ului.
- `frames` : reprezintă o listă de containere text sau imagini care la rândul lor au următoarele chei: `"id"`, `"type"`, `"content"`, `"x"`, `"y"`, `"width"` și `"height"`.

Un exemplu cu aceasta mapare este în Figura 4.3.

```
1 const initialSlidesList=[
2   {
3     id : 0,
4     background: 'white',
5     frames: [
6       {
7         id : 0,
8         type : 1,
9         content : "",
10        x : 0,
11        y : 0,
12        width : '150px',
13        height : '150px',
14      }
15    ]}]
16 const [slidesList, setSlideList] = useState(initialSlidesList);
```

Figura 4.3: Structura de date folosită în partea de front-end

Pagina este împărțită în 3 componente mari: meniul, bara de intrumente și zona de editare.

4.3.1 Meniul

Acest component include: logoul aplicației FrameIT, câmpul pentru setarea titlului proiectului, zona pentru eventuale erori, zona de vizualizare a slide-urilor proiectului, butonul de adăugare slide "New Slide" și butonul de salvare "Save".

Zona de vizualizare a slide-urilor proiectului reprezintă un container scrolabil vertical, în care sunt stocate slide-urile prezentării într-o variantă mică, de previzualizare. Slide-urile sunt niște containere de mărime fixă cu raportul de aspect 16:9 care se crează în mod dinamic prin iterarea listei de slide-uri din Figura 4.3. Din această iterare obținem id-ul slide-ului și culoarea fundalului. Ulterior iterăm și prin containerele fiecărui slide, numite "frames". În timpul iterației verificăm tipul contain-urilor pentru a putea face diferență între text și imagine, deoarece construcția lor diferă. Din această iterație obținem toate datele necesare pentru construirea containerelor de tip text și de tip imagine. Însă, din cauza că aceste containere au fost create în zona de editare a unui slide curent, rezultă că și valorile datelor sale sunt mult prea mari pentru a le include în previzualizarea slide-urilor, de aceea am folosit funcționalitatea din CSS: `transform:scale(0.1)`. Astfel dimensiunile și valorile poziției fiecărui container au fost scalate cu 0.1 și au ajuns la dimensiunea și poziția necesară pentru a le include în interiorul previzualizării slide-ului.

Containerele de text pentru slide-urile de previzualizare sunt create cu ajutorul librăriei externe React-Quill, ce reprezintă un editor de text bogat. Avem nevoie de această librărie pentru că datele din "content" sunt în format HTML, iar React-Quill este un editor HTML și poate traduce codul HTML fără să piarda din stilul, culoarea și dimensiunea textului. În această situație nu avem nevoie de această librărie pentru a edita textul, de aceea setăm parametrul "ReadOnly" pe adevărat pentru containerele noastre pentru a putea doar vizualiza conținutul.

Imaginile la randul lor la fel sunt construite cu ajutorul unui container, căruia i se setează toți parametri necesari: dimensiunea, poziția, iar imaginea este setată ca fundal al acestui container cu ajutorul funcționalităților din CSS:

```
1 backgroundImage:url(`${frame.content}`),  
2 backgroundSize:cover,  
3 backgroundRepeat:no-repeat
```

Figura 4.4: Funcționalități CSS utilizate pentru afișarea imaginilor

Astfel, imaginea acoperă întreg containerul fără ca să se repete sau să își strice calitatea. Pe lângă această structură de date folosite pentru ca aplicația să înțeleagă ce slide trebuie afișat în zona de editare se mai folosește și o variabilă ce stochează id slide-ului curent, numită "currSlideId". La selectarea unui slide din lista de slide-uri de previzualizare se modifică și variabila "currSlideId" pentru a ști ce slide să afișeze în zona de editare.

```
1
2  const [currSlideId, setCurrSlideId]=useState(initialSlidesList.length>0?
   initialSlidesList[0].id : null);
```

Figura 4.5: Variabilă utilizată pentru afișarea slide-ului în zona de editare

Butonul "New Slide" crează un slide nou, folosind niște parametri presetati, pe care îl adaugă în lista de slide-uri, și actualizează variabila "currSlideId" cu valoarea id-ului slide-ului selectat. Astfel, în zona de editare se va afișa ultimul slide adăgat.

Butonul "Save" structurează toate informațiile într-un dicționar numit "inputs" ce are următoarele chei :

- "projectId" - la editarea unui proiect stochează id-ul proiectului curent, la crearea unui proiect nou va fi null,
- "projectName" - stochează numele proiectului,
- "slidesList" - stochează lista de slide-uri,
- "userId" - stochează id utilizatorului extras din memoria locală.

Acum prin apelurile de API:

```
1 // folosit pentru editarea proiectelor
2 axios.post( /editProject ,inputs,{withCredentials:true, })
3 // folosit pentru crearea proiectelor
4 axios.post( /createProject ,inputs,{withCredentials:true, })
```

se transmit datele către server unde se analizează informația și se returnează un mesaj de validare sau un mesaj de eroare.

În server, se decriptează tokenul de acces pentru a verifica identitatea utilizatorului, se compară id-ul primit de la client cu cel salvat în tokenul de acces. Dacă identitatea este confirmată, folosind interogarea:

```
1 "SELECT * FROM projects WHERE name=? AND authorId=?"
```

se verifică dacă acest utilizator mai are în baza de date un proiect cu astfel de titlu. În cazul în care există, la crearea unui proiect nou serverul va trimite înapoi un mesaj de eroare, altfel validează și inserează datele în baza de date. Însă, la editarea unui proiect existent serverul verifică dacă id-ul returnat de interogare diferă de id-ul proiectului, în cazul în care id-ul diferă înseamnă că în baza de date utilizatorul are un alt proiect cu titlul dat și serverul va returna un mesaj de eroare către client, altfel va returna un mesaj de validare și va face actualizările în baza de date.

4.3.2 Bara de instrumente

Bara de instrumente este împărțită în două componente: componenta de inserare și editare a containerelor text și imagini, numite "frames" și componenta de editare textului. Pentru a folosi din plin acest două elemente, aplicația folosește încă o variabilă "currFrameId", în care se stochează id-ul containerului de tip text sau imagine curent.

```
1  const [currFrameId, setCurrFrameId]=useState();
```

- Componenta de inserare și editare a containerelor

Această componentă conține următoarele funcționalități:

- Adăugarea text : La adăugarea containerului de text se va crea un nou container care se va adăuga în lista de containere, numite "frames" ale slide-ului curent. La fel se actualizează variabila "currFrameId" cu valoarea id-ului ultimului container adăugat. La final, containerul apare în colțul stâng-sus al slide-ului curent.
- Adăugarea imagine : La adăugarea unei imagini, imaginea se convertește într-un șir de caractere în baza 64, apoi se va crea un nou container de tip 2 care se va adăuga în lista de containere, numite "frames" ale slide-ului curent. La fel se actualizează variabila "currFrameId" cu valoarea id-ului ultimului container adăugat. La final, imaginea apare în colțul stâng-sus al slide-ului curent.
- Ștergerea containerului : În urma ștergerii containerului curent de tip text sau de tip imagine din cadrul slide-ului curent. Variabila "currFrameId" devine null și se elimină acest container din lista de containere "frames" al slide-ului curent.
- Editarea slide-ului : Accesarea editării slide-ului setează variabila booleană "showEditSlidePopUp" pe adevărat. Acest lucru schimbă stilul la câteva contai-

nere din această pagina pentru a nu se suprapune conținutul ferestrei de editare a slide-ului cu celelalte elemente ale paginii. De exemplu, poziția slide-ului curent din zona de editare din relativă devine statică, iar containerele din interiorul slide-ului cu totul dispar în timpul când această fereastră este activă. Această pagină permite setarea și modificarea nuanței fundalului slide-ului curent în timp real, datorită faptului că odată ce este apăsat butonul "save" se verifică dacă culoarea a fost modificată, în caz afirmativ se actualizează elementul curent din lista de slide-uri. Pentru elementul de selectare a culorii am folosit librăria React-Color, mai exact containerul BlockPicker.

- Componenta de editare text

Bara de instrumente este o componentă esențială într-o aplicație de editare a textului. Pentru editarea textului a fost creată de la zero o bara de instrumente proprie, deoarece ceea oferită de librăria React-Quill nu a putut fi utilizată în contextul aplicației FrameIT. Librăria externă oferă o bară nedetașabilă de containerul de editor de text. Însă aplicația web FrameIT are nevoie de o bară de instrumente separată pentru a o poziționa cât mai intuitiv și convenabil în pagina și pentru a edita mai multe containere de tip text folosind o singură bară, mai mult se dorea ca utilizatorul să vizualizeze direct în slide-ul curent schimbările efectuate în text. Pentru a putea edita separat mai multe containere de tip text din același slide a fost nevoie de utilizarea unei liste de referințe a containerelor din cadrul slide-ului. Astfel, bara de instrumente știe ce container o accesează.

```
1  const editorRefs = useRef({});
```

A fost creată o funcție care se ocupă de comunicarea dintre bara de instrumente FrameIT și librăria React-Quill (Figura 4.6). Astfel, ea păstrează majoritatea funcționalităților de formatare și editare a textului din bara librăriei.

Funcția la început verifică dacă lista de slide-uri există și dacă tipul containerului din cadrul slide-ului este 1, adică de tip text. Ulterior se focalizează pe editorul de text corespunzător componentei curente și se obține obiectul Quill. Mai departe, funcția verifică dacă formatul cerut de utilizator este de tip: bold, italic sau underline. În caz afirmativ, se verifică formatul curent al textului, Dacă textul nu are formatul dat, atunci i se atribuie, în caz contrar textului i se elimină formatul cerut de utilizator. Dacă este alt format decât cele enumerate mai sus, atunci se

obține selecția curentă din editor și i se atribuie formatul. La final, se restabilește selecția.

```
1 const handleFormat = (format, value = true) => {
2   if(slidesList&& slidesList[currSlideId-1].frames[currFrameId-1].type===1)
3     editorRef.current[currFrameId-1].focus();
4   const quill = editorRef.current[currFrameId-1].getEditor();
5   if(format==='bold' || format==='italic' || format==='underline'){
6     const isFormat=quill.getFormat()[format];
7     quill.format(format,!isFormat);
8   }else{
9     const range = quill.getSelection();
10    quill.format(format, value);
11    quill.setSelection(range);
12  }}
```

Figura 4.6: Funcția asigură comunicarea dintre bara de instrumente și librăria React-Quill

4.3.3 Zona de editare

Zona de editare reprezintă un container în care se găsește slide-ul curent cu toate containerele sale de tip text și imagini. Slide-ul la rândul său reprezintă un container de o dimensiune fixă cu raportul de aspect 16:9. Datorita variabilei "currSlideId", care stochează id-ul slide-ului curent, aplicația FrameIT înțelege care slide trebuie afișat în zona de editare.

```
1 <div className="slide"
2     style={{background:slidesList[currSlideId-1].background}}>
3   {currSlideFrames}
4 </div>
```

Figura 4.7: Afișarea slide-ului curent în zona de editare.

Afișarea containerelor de tip text și imaginile se face în mod dinamic, prin iterarea containerelor slide-ului current, numite "frames". În primul rând se verifică tipul lor, deoarece metodele de creare sunt diferite.

Pentru construcția containerelor de tip text sunt folosite două librării externe: React-Rnd și React-Quill. Funcționalitatea de redimensionare și mutare a componentelor de tip text în interiorul slide-ului se realizează prin intermediul containerelor Rnd. În cazul glisării, pentru a ști poziția nouă a unui container, la finalul mutării se actualizează datele "x" și "y" ale acestui element. În cazul redimensionării, la finalul acțiunii se actualizează datele "width" și "height" ale containerului. Pentru ca containerele să nu poată fi glisate înafara slide-ului se utilizează proprietatea: "bounds='parent'", unde parintele este slide-ul curent. Funcționalitatea de editare a textului din interiorul containerului este garantată de librăria React-Quill. În interiorul containerului Rnd se construiește containerul ReactQuill, care reprezintă

```

1 var currSlideFrames=currSlideId && slidesList[currSlideId-1].frames.map((
    frame)=>{
2   return (
3     <Rnd className={`ql-container ${currFrameId === frame.id ?
                                     'activeFrame' : ''}`}
4       key={key}; bounds="parent" ;
5       position={{ x: frame.x, y: frame.y }} ;
6       enableResizing=true;
7       size={{ width: frame.width, height: frame.height }};
8       onClick={()=>handleFrameClick(frame.id)};
9       onDragStop={(e,d)=>{draggableEventHandler(d.lastX,d.lastY,...)}};
10      onResizeStop={(e,direction,ref,delta,position)=>{
                                     resizebleEventHandler(ref,frame.id,position);}}>
        <ReactQuill
11          ref={(element) => editorRefs.current[frame.id-1]=element};
12          value={frame.content};
13          onChange={(props)=>handleEditorChange(props,frame.id)};
14          placeholder="Text";
15          modules={{ toolbar: false }};/>
16    </Rnd> ) }

```

Figura 4.8: Afișarea containerelor de tip text în cadrul slide-ului curent.

Pentru construcția containerelor de tip imagine se folosește componenta librăriei React-Rnd cu aceleași proprietăți și parametri ca și la construcția containerelor de tip text, doar că fundalul deja nu mai este transparent dar este imaginea dorită. Acest lucru este realizat folosind proprietățile din Figura 4.4

Concluzii

În cadrul lucrării pentru teza de licență am prezentat dezvoltarea unei platforme, numită FrameIT, dar și crearea acesteia de la zero. Platforma reprezintă o bună oportunitate de a crea prezentări. Utilizatorii pot crea prezentări folosind funcții simple, cu un design minimalist, dar interesant pentru a atrage atenția. Pentru a începe proiectul a fost nevoie de o cercetare cu scopul de a identifica corect cerințele utilizatorilor.

Dezvoltarea platformei a început cu selectarea funcțiilor de bază ușor de utilizat și eliminarea tuturor complexității. A fost concepută o interfață atractivă, care să permită celor care o utilizează să creeze conținut, să adauge și să editeze textul, imaginile. Platforma oferă și posibilitatea de stocare și gestionare a prezentărilor. Totodată oferă și opțiuni pentru a personaliza prezentarea, aspectul poate fi adaptat după preferințe, alegerea oricărei game cromatice.

Tehnologiile utilizate în dezvoltarea aplicației FrameIT au satisfăcut în totalitate cerințele aplicației și mi-au oferit oportunitatea de a crește și de a evolua ca programator. Pe măsură ce am lucrat cu aceste tehnologii în timpul proiectului, am dobândit abilități importante, pregătindu-mă pentru a aborda și alte proiecte web complexe în viitor.

Ca și orice altă platformă, FrameIT va trece printr-o îmbunătățire și dezvoltare continuă în viitor. Cu timpul vor fi incluse mai multe funcționalități, la fel de simple pentru a putea fi în continuare utilizată de către oricine. Odată ce platforma va începe să fie utilizată va apărea și un feedback, prin intermediul căruia vor apărea și anumite sugestii din partea utilizatorilor, fapt ce va contribui la dezvoltarea platformei, devenind mai performantă.

Platforma Frame IT reprezintă o soluție inovatoare pentru prezentări simple, ușor de creat și de editat. Ea oferă utilizatorilor posibilitatea de a crea și oferi prezentări personalizate. Viitorul platformei va aduce mai multe funcționalități și îmbunătățiri pentru a satisface nevoile în continuă evoluție.

Bibliografie

- Author1, *Book1*, 2018
- Author2, *Boook2*, 2017