**SPIDER**

**PROIECTAREA ȘI IMPLEMENTAREA UNUI SISTEM DE MANAGEMENT BAZAT PE METODA KANBAN**

**Candidat: David-Marian BOTEZATU**

**Coordonator științific: Ș.l.dr.ing. Mihaela Marcella CRIȘAN-VIDA**

Sesiunea: Iunie 2023

# INTRODUCERE

## CONTEXT

## TEMA LUCRĂRII DE LICENȚĂ

# STUDIU BIBLIOGRAFIC

# FUNDAMENTARE TEORETICĂ

Acest capitol are ca scop prezentarea tehnologilor și bibliotecilor folosite în crearea aplicației descrise în [Tema lucrării de licență](#_TEMA_LUCRĂRII_DE). Componentele principale în crearea aplicației client sunt Axios și React, iar pentru server folosim NodeJS împreună cu Express. Pentru baza de date folosim PostgreSQL, iar pentru cache și token management folosim Redis. În continuare, vom detalia aceste tehnologii și biblioteci pentru a motiva alegerea lor în crearea aplicației. Totusi, deoarece Spider este o aplicație de tip client-server, și tehnologiile și bibliotecile folosite vor fi împărțite în aceleași categorii, prima categorie fiind reprezentata de catre bibliotecile folosite in crearea aplicatiei client:

1. **Axios:** conform site-ului oficial, Axios este un client HTTP pentru browser și node.js[[1]](#footnote-1). Prin urmare, Axios este o bibliotecă JavaScript open-source, folosită pentru a realiza cereri HTTP din browser către un server Node.js, sau mai exact, către un API care rulează într-un mediu Node.js. Spider folosește Axios doar în aplicația client pentru a face cereri și a trimite date către API. Am ales această bibliotecă în detrimentul altora precum Fetch deoarece este mai simplu de utilizat si mai sigur (avand integrata protectie anti XSRF)
2. **React**: este o bibliotecă JavaScript dezvoltată de Meta, folosită în construirea interfețelor de utilizator (UI) pentru aplicații web și aplicații mobile[[2]](#footnote-2). React folosește bucăți individuale de cod în construcția acestor interfețe, numite **componente**. Acestea pot fi scrise de către un programator, sau pot exista în alte librării, și pot fi apelate doar atunci când este nevoie de ele și reprezintă una din principalele caracteristici ale React. Astfel, aceste componente sunt reutilizabile, ceea ce reprezintă un avantaj major deoarece facilitează organizarea și structurarea codului și duc spre crearea de aplicații modulare și flexibile. În crearea acestor componente React permite dezvoltatorilor să folosească o extensie a JavaScript numită JSX (JavaScript Syntax Extension), prin care aceștia pot adăuga cod HTML (sau alte componente/elemente React) în variabile sau metode JavaScript[[3]](#footnote-3). O altă caracteristică este reprezentată de procesul de randare: React folosește o tehnică numită **Virtual DOM** pentru actualizarea și reprezentarea aplicațiilor. Prin această tehnică React actualizează doar acele părți din aplicație care necesită schimbate în loc să actualizeze toată pagina web la fiecare modificare, acest proces permițând o actualizare mai rapidă și eficientă a interfeței[[4]](#footnote-4). Bineînțeles, există mai multe librării care permit dezvoltarea de aplicații web precum Vue.js, Angular sau Ember.js, dar am ales datorită caracteristicilor descrise mai sus, plus simplitatea acestuia și comunitatea și ecosistemul bogat in biblioteci și instrumente care ușurează și îmbunătățesc procesul de dezvoltare al aplicațiilor.
3. **React-beautiful-dnd**: este una din bibliotecile React folosite pentru gestionarea funcțiilor de tip drag-and-drop. În aplicația Spider este folosită în gestionarea tabelei Kanban prin crearea a două componente principale: Kanban Column și Kanban Task, în care Kanban Column este componenta statică, cea care nu poate fi mutată, iar Kanban Task este componenta care poate fi mutată și plasată în interiorul altor coloane. Deși există mai multe biblioteci care oferă acest tip de funcționalitate, am ales această bibliotecă deoarece este simplu de folosit si in acelasi timp este si foarte puternica, oferind o performanta ridicata si flexibilitate in personalizare.
4. **React-hook-form**: este o biblioteca simpla, flexibila si performanta, folosita pentru gestionarea formularelor in aplicatiile React[[5]](#footnote-5). Desi exista si alte biblioteci care ajuta la gestionarea formularelor (precum Formik), am ales aceasta biblioteca doarece este integrata direct cu React, ofera o performanta ridicata comparativ cu celelalte biblioteci si este usor de utilizat[[6]](#footnote-6).
5. **React-icons**: aceasta biblioteca ofera o solutie usoara si convenabila pentru utilizarea pictogramelor in aplicatii React. Am ales aceasta solutie deoarece ofera o gama larga de seturi de pictograme care pot fi utilizate printr-un simplu import in aplicatie. De asemenea, aceste pictograme pot fi usor personalizate folosind CSS si au o performanta optimizata pentru a asigura o incarcare rapida si o experienta placuta pentru utilizatori.
6. **React-modal**: este o bibliotecă React utilizată pentru a crea și gestiona ferestre modale (pop-up) în aplicațiile web intr-o maniera simplificata, oferind un control flexibil si complet asupra afisarii si utilizarii acestora. Am ales aceasta biblioteca deoarece este usor de utilizat - ofera o componenta React care poate fi personalizata folosind diferite proprietati create in React si este usor de integrat cu alte componente sau biblioteci UI precum Tailwind Css pentru personalizare vizuala.
7. **React-router-dom**: o bibliotecă esențială folosita pentru gestionarea rutelor și navigării pe partea de client (client side routing) în aplicațiile web bazate pe React, prin care dezvoltatorii pot defini configuratia rutelor intr-un mod declarativ utilizand componente React. Acest stil de rutare permite aplicatiei sa schimbe un URL fara a cere un document nou de la server iar in acest fel, aplicatia poate sa afiseze imediat o componenta noua.
8. **React-switch**: este o bibliotecă React utilizată pentru crearea și gestionarea comutatoarelor în aplicațiile web. Ofera o componenta <Switch> care poate fi customizata utilizand CSS (sau o biblioteca UI precum Tailwind).
9. **React-toastify**: este utilizata pentru crearea si gestionarea notificarilor dintr-o aplicatie React pe baza componentelor. Am preferat aceasta solutie deoarece este usor de utilizat, permite personalizarea aspectului si comportamentului notificarilor (precum culoare, pozitionare, timp de afisare, etc). De asemenea, pot fi setate sa fie interactive, dezvoltatorii putand adauga butoane care executa diferite actiuni in functie de tipul de notificare.
10. **Yup**: este o biblioteca folosita pentru a aplica validari pe datele de intrare din formulare folosind o schema definita de dezvoltator. De asemenea, dezvoltatorii pot crea mesaje de eroare personalizate care apoi sunt afisate in aplicatie.
11. **TailwindCSS**:

Pentru aplicatia server si API am folosit urmatoarele biblioteci si tehnologii:

1. NodeJS
2. Bcryptjs
3. Body-parser
4. Cors
5. Dotenv
6. ExpressJS
7. Generate-password
8. Ioredis
9. Jsonwebtoken
10. Nodemailer
11. Nodemon
12. Pg
13. Pg-hstore
14. Sequelize

# SPECIFICAȚIILE APLICAȚIEI

## DESCRIEREA APLICAȚIEI

## FUNCȚIILE APLICAȚIEI

## INTERFAȚA CU UTILIZATORUL

## BAZA DE DATE

## CONECTAREA CU ALTE APLICAȚII SAU SISTEME

# PROIECTAREA APLICAȚIEI

## COMPONENTELE APLICAȚIEI

## STOCAREA INFORMAȚIEI

# IMPLEMENTAREA APLICAȚIEI

# MANUAL DE UTILIZARE

# CONCLUZII ȘI DIRECȚII VIITOARE

Fiecare capitol trebuie să aibă o structură clară, va începe pe pagină nouă și va conține un titlu. Va fi urmat de două linii de 12 pt lăsate libere.

Fiecare secțiune a unui capitol (ex. 1.1 INFORMAȚII GENERALE) va fi poziționat la un rând liber sub text și va avea un rând liber de12 pt deasupra textului.

Textul lucrării va fi aliniat uniform (justify). Este de preferat ca textul să fie verificat pentru eventualele erori în limba de editare cu ajutorul facilității de verificare a ortografiei (speller) din programul word. Este recomandat ca lucrarea de finalizare a studiilor să nu depășească 100 de pagini, inclusiv anexele.

Reguli aplicate pentru textul lucrării:

a. Marginile paginii – se vor utiliza următoarele valori pentru marginile paginii (Page Setup -> Margins-> Mirror Margins):

 interior: 2 cm

 exterior: 2 cm

 sus: 2,5 cm (inclusiv header)

 jos: 2 cm

b. Spațiere între rânduri - textul va respecta o spațiere între rânduri de 1,15 linii (Format>Paragraph->Line spacing-> 1,15 lines);

c. Alinierea textului în cadrul paragrafelor - textul din cadrul paragrafelor normale va fi aliniat între marginile din stânga şi dreapta (justified). Primul rând al fiecărui paragraf va avea o aliniere de 1,5 cm (Format-> Paragraph-> Indentation-> Left). Excepție fac titlurile capitolelor, care vor fi aliniate la stânga, precum și etichetele tabelelor și ale figurilor (conform explicațiilor de mai jos);

d. Font – fontul utilizat pentru redactare va fi Arial, cu dimensiunea de 12 puncte, utilizând diacriticele specifice limbii în care este redactată lucrarea (ex: ă, ş, ţ, î, â - pentru limba română);

e. Numerotarea paginilor - numerotarea paginilor se face începând cu pagina de titlu, până la ultima pagină a lucrării, dar numărul paginii apare doar începând cu Introducerea. Numărul de pagină se inserează în subsolul paginii, centrat.

f. Antetul paginii – apare începând cu introducerea și va conține pe rânduri succesive un text cu înălțimea de 8, aliniat la stânga: (i) textul Universitatea Politehnica Timișoara ; (ii) denumirea programului de studii și anul susținerii ; (iii) numele candidatului (în stânga) și titlul lucrării. În partea dreaptă a antetului poate fi integrată sigla UPT;

## FIGURI ȘI FOTOGRAFII (12 pt, Bold, Uppercase, Left)

Figurile (incluzând imagini, grafice, capturi de ecran) se numerotează în ordinea apariției în lucrare. Alternativ, figurile pot fi numerotate în ordine în fiecare capitol, integrând în numerotare și numărul capitolului. Fiecare figură are număr și titlu, care se menționează sub figură, centrat. Dacă este cazul, sursa figurii se indică între paranteze după titlul figurii. Toate figurile şi fotografiile prezentate în lucrare trebuie să fie referite în textul lucrării, trebuie să fie numerotate şi însoţite de titlul figurii.

Se va lăsa câte o linie liberă (Arial 12 pt) între figură şi text. Figurile vor fi centrate pe pagină.

INT_EXTjoint

Figura 1 – Exemplu de figură (sursa: Buletinul ştiinţific al UPT seria construcţii-arhitectură nr.2 /2010)

## TABELE (12 pt, Bold, Uppercase, Left)

Tabele – tabelele se numerotează în ordinea apariției în lucrare. Alternativ, tabelele pot fi numerotate în ordine în fiecare capitol, integrând în numerotare și numărul capitolului. Fiecare tabel are număr și titlu, care se menționează deasupra tabelului, aliniat centrat. Dacă este cazul, sursa datelor se precizează între paranteze după titlul tabelului;

Toate tabelele prezentate în lucrare trebuie să fie referite în textul lucrării, trebuie să fie numerotate şi însoţite de un titlu (vezi exemplul de mai jos). Dacă se utilizează figuri copiate atunci se va indica sursa fotografiei în paranteză. Pe cât posibil, în tabel se va păstra fontul uzual (arial 12 pt) dar sunt acceptate şi modalităţi de a scoate în evidenţă rezultatele importante (bold, italic etc.)

Se va lăsa câte o linie liberă (arial 12 pt) între text şi tabel. Tabelele vor fi centrate pe pagină.

Tabelul 1. Exemplu de tabel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Yield stress, fy [N/mm2] | | Tensile strength, fu [N/mm2] | |
| Element | Mill certificate | Coupon tests | Mill certificate | Coupon tests |
| Beam IPE360 | 285.0 | 329.8 flange  348.4 web | 427.0 | 463.2 flange  464.0 web |
| Column HEB300 | 311.3 | 313.0 flange  341.8 web | 446.0 | 449.8 flange  464.4 web |
| End plate | 281.0 | 248.3 | 424.7 | 416.0 |
| Cover plate | 296.0 | 273.2 | 443.0 | 436.7 |

## FORMULELE (12 pt, Bold, Uppercase, Left)

Formulele utilizate în text se vor numerota în ordinea apariției în lucrare. Alternativ, formulele pot fi numerotate în ordine în fiecare capitol, integrând în numerotare și numărul capitolului. Numerotarea formulelor se face în paranteze rotunde. Se va lăsa câte o linie liberă (arial 12 pt) între text şi formulă. Formulele vor fi aliniate la dreapta.

(1)

# N. CONCLUZII (14 pt, Bold, Uppercase, Center)

Lucrarea se va încheia cu un capitol de concluzii. Acesta va conţine principalele rezultate ale lucrării şi implicaţiile practice ale acestora. În cazul proiectelor de diplomă, se vor menționa principalele date sintetice obținute din procesul de proiectare

# BIBLIOGRAFIE

La sfârşitul lucrării va fi dată o listă de referinţe pentru textele ştiinţifice consultate pe parcursul realizării lucrării. Vor fi trecute toate sursele, inclusiv cele de pe internet. Acestea vor fi referite în text şi trecute în lista de referinţe în ordine alfabetică, după exemplele de mai jos.

Bibliografia trebuie să cuprindă toate titlurile din literatura de specialitate care au servit ca bază de documentare, respectiv autorii care au fost citați în text, la toate capitolele lucrării.

În cadrul Facultății de Automatică și Calculatoare se cere folosirea stilului de citare IEEE (detalii [IEEE Citation Guidelines2.doc (ieee-dataport.org)](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf)), folosit cu precădere în publicațiile științifice din domeniul IT. Cele trei părți importante ale referinței sunt:

* Numele autorului indicat ca prima inițială a prenumelui, apoi numele complet.
* Titlul articolului, brevetul, lucrarea de conferință etc., între ghilimele.
* Titlul revistei sau cărții cu caractere cursive

Modul de redactare a referinței depinde de tipul publicației, vă rugăm să urmăriți cu atenție indicațiile de la link-ul de mai sus.

Fiecare citare trebuie notă în text prin utilizarea unor numere secvențiale simple. Un număr cuprins între paranteze drepte, plasat în textul raportului, indică referința specifică. Citările sunt numerotate în ordinea în care apar. Odată ce o sursă a fost citată, același număr este folosit în toate referințele ulterioare din text. Nu se face distincție între sursele electronice și cele tipărite, cu excepția detaliilor referințelor citate.

Fiecare număr de referință trebuie să fie cuprins între paranteze drepte pe aceeași linie cu textul, înaintea oricărei semne de punctuație, cu un spațiu înaintea parantezei.

Exemple

„. . .finalul cercetării mele [13].”

„Teoria a fost prezentată pentru prima dată în 1987 [1].

Lista de referințe din bibliografie este compusă din toate sursele folosite pentru documentarea lucrării și se realizează în ordinea numerică a citării în text și nu în ordine alfabetică a autorilor.

Preluarea identică a unei fraze sau paragraf va fi citată prin indicarea inclusiv a paginii din sursa utilizată, dar și prin ghilimele şi forma italică a literelor; pentru sursele preluate de pe internet, vor fi notate adresele de pagină web; în lista bibliografică finală lucrările se trec în ordinea alfabetică a numelor autorilor. La lucrările colective, regula referitoare la ordinea alfabetică este valabilă pentru primul autor.

Dacă se citează site-uri web, reviste sau articole, înainte de acestea se vor trece trei asteriscuri, informații referitoare la volum, număr, pagini consultate, adresa web exactă a articolului respectiv, data vizitării site-ului și a descărcării materialului, data accesării. Adresele de pagini web se regăsesc la finalul listei.

Sursele bibliografice la care nu se poate menționa autorul se vor specifica astfel: „\*\*\*”urmat de denumirea articolului și/sau a cărții, editura și locul apariției (pentru cărți), volumul, numărul acestuia, prima și ultima pagină a lucrării citate, anul apariției.

\*\*\* https://ro.wikipedia.org/wiki/Motor\_cu\_reac%C8%9Bie accesare februarie 2022

Ultima pagină a lucrării de licență/diplomă/disertație trebuie să conțină „Declarația de originalitate a lucrării de finalizare a studiilor”, completată olograf, în conformitate cu cerinţele UPT. Declaraţia se descarcă de pe adresa de web:

[DECLARAŢIE DE AUTENTICITATE A (upt.ro)](http://www.upt.ro/img/files/Regulamente_UPT/2020/Declaratie_de_autenticitate_UPT_2020.pdf)

1. Website-ul Axios, iunie 2023: https://axios-http.com/docs/intro [↑](#footnote-ref-1)
2. Website-ul React, iunie 2023: https://react.dev/ [↑](#footnote-ref-2)
3. Wikipedia, iunie 2023: https://en.wikipedia.org/wiki/JSX\_(JavaScript) [↑](#footnote-ref-3)
4. Website-ul React legacy, iunie 2023: https://legacy.reactjs.org/docs/faq-internals.html [↑](#footnote-ref-4)
5. Website-ul oficial react-hook-form, iunie 2023: https://www.react-hook-form.com/ [↑](#footnote-ref-5)
6. https://www.react-hook-form.com/faqs/#ReactHookFormFormikorReduxForm [↑](#footnote-ref-6)