Interfața aplicatiei orar a Universității „Ștefan cel Mare” – Suceava

-Proiect de diplomă-

Profesor îndrumător: Absolvent:

conf. dr. ing. Schipor Ovidiu-Andrei  Ailoaie Cosmin Iulian

Suceava 2020

Cuprins

[Tabel de figuri 3](#_Toc45081600)

[Introducere 4](#_Toc45081601)

[I.Tehnologii utilizate 6](#_Toc45081602)

[I.1.VueJS 6](#_Toc45081603)

[I.2.TypeScript 10](#_Toc45081604)

[I.3.Alte pachete și tehnologii 11](#_Toc45081605)

[I.3.1.BootstrapVue și Bootstrap 11](#_Toc45081606)

[I.3.2. HTML și SCSS (Sassy Cascading Style Sheets ) 12](#_Toc45081607)

[II.Proiectarea și implementarea 14](#_Toc45081608)

[II.1.Descriere generală 14](#_Toc45081609)

[II.2.Lista componentelor și structura acestora 16](#_Toc45081610)

[III.Utilizarea aplicației 18](#_Toc45081611)

[III.1.Prima pagină 18](#_Toc45081612)

[III.2.Modul Day/Night 20](#_Toc45081613)

[III.3.Orare favorite 21](#_Toc45081614)

[III.4.Facultăți 22](#_Toc45081615)

[III.5.Profesori / Cadre didactice 24](#_Toc45081616)

[III.6. Orarul 26](#_Toc45081617)

[IV.Implementări ulterioare 28](#_Toc45081618)

[Concluzii 30](#_Toc45081619)

[Bibliografie 31](#_Toc45081620)

# Tabel de figuri

[Fig. I. 1 Exemplu componentă Vue 6](#_Toc45078708)

[Fig. I. 2 Exemplu de reutilizare a componentelor Vue 7](#_Toc45078709)

[Fig. I. 3 Element cuprins într-un tag transition 7](#_Toc45078710)

[Fig. I. 4 Aplicarea unui efec de tranziție care durează 0.5 secunde 8](#_Toc45078711)

[Fig. I. 5 Integrarea automată intre JavaScript și HTML oferită de VueJS 8](#_Toc45078712)

[Fig. I. 6 Exemplificarea directive v-for din VueJS 9](#_Toc45078713)

[Fig. I. 7 Componentă VueJS creată sub formă de folosind TypeScript 10](#_Toc45078714)

[Fig. I. 8 Exemplu de interfată în TypeScript pentru VueJS 11](#_Toc45078715)

[Fig. I. 9 Implementarea BootstrapVue a claselor în tag-uri 12](#_Toc45078716)

[Fig. I. 10 Șablon document HTML 12](#_Toc45078717)

[Fig. I. 11 Exemplu de cod SCSS și echivalentul său în CSS 13](#_Toc45078718)

[Fig. II. 1 - Interfața Store-ului și starea inițială 14](#_Toc45078646)

[Fig. II. 2 - Exemplu de apel cadru didactic și răspunsul cu activitățile 15](#_Toc45078647)

[Fig. II. 3 - Executarea unei acțiuni care resetează conținutul store-ului 15](#_Toc45078648)

[Fig. II. 4- Rutele în interiorul aplicației și calea acestora 16](#_Toc45078649)

[Fig. II. 5 Folosirea v-for și a unei componente Vue pentru afișarea informațiilor pentru activități 17](#_Toc45078650)

[Fig. III. 1 - Prima pagină. varianta în modul Day 18](#_Toc45078598)

[Fig. III. 2 - Bara de navigare deschisă, modul Day 19](#_Toc45078599)

[Fig. III. 3 - Vedere mobile pe un iPhone SE 19](#_Toc45078600)

[Fig. III. 4 - Exemplificare intre modurile Day - Night, vedere mobile 20](#_Toc45078601)

[Fig. III. 5 - Listă orare favorite și listă goală 21](#_Toc45078602)

[Fig. III. 6 - Lista facultăților disponibile 22](#_Toc45078603)

[Fig. III. 7 - Lista specializărilor și grupele pentru Calculatoare, FIESC. 23](#_Toc45078604)

[Fig. III. 8 - Lista celor 570 de cadre didactice. 24](#_Toc45078605)

[Fig. III. 9 - Lista celor 570 de cadre didactice, vedere mobile. 25](#_Toc45078606)

[Fig. III. 10 - Activitățile unui cadru didactic și tooltip pentru informații suprmentare. 26](#_Toc45078607)

[Fig. III. 11 - Orar favorit, versiunea mobile, pentru ziua de luni. 27](#_Toc45078608)

# Introducere

Prima impresie este foarte importantă, de aceea, toate aplicațiile, site-urile web și produsele care urmează să fie folosite de utilizatori, trebuie să aibă și o interfață plăcută din punct de vedere vizual, intuitivă și ușor de folosit.

Într-o explicație simpla, UI ( user interface ) reprezintă o serie de ecrane, pagini și elemente vizuale, cum ar fi butoanele și imaginile, care ajută o persoană sa interacționeze cu un produs sau serviciu.

Mai există și conceptul de UX ( user experience ) care este interpretat ca și experiența, fie ea pozitivă, negativă sau neutră, pe care utilizatorul o resimte interacționând cu interfața produsului pe care îl folosește.

Astfel, am dorit ca o parte importantă din viața de student și anume, activitățile desfășurate în cadrul universității, iar orarul activaților să fie mai ușor de accesat și de înțeles pe platforma USV. De aceea am ales ca interfața aplicației „Orar USV” să primească o bine meritată actualizare, folosind VueJs, care este un framework open-source nou, lansat în 2014 [1] și foarte popular [2] pentru partea de front-end în crearea de SPA-uri ( single-page aplication ) cu un accent pus pe viteză și ușurință în folosire și care m-a ajutat să îmbunătățesc interfața, și astfel experiența pe care studenții o vor avea în folosirea aplicației, fără ca partea de back-end să necesite modificări pentru moment.

Deoarece orarul Universității „Ștefan cel Mare” – Suceava este folosit de către toți studenții și de către cadrele didactice, are nevoie de o interfață prietenoasă, actualizată, de ce nu, la WEB 3.0, pentru o experiență mai plăcută pe partea de desktop, dar mai ales pe partea de mobile, unde interfața actuală suferă cel mai mult și este foarte greu de folosit, iar numărul accesărilor este tot mai mare pentru că și numărul dispozitivelor mobile a crescut foarte mult. Astfel, interfața nouă este adaptivă, axată pe ușurință în utilizare pe partea de dispozitive mobile, cu ecran și rezoluție mică.

În același timp, un motiv important în alegerea acestei teme a fost de a lăsa actualilor, dar mai ales viitorilor studenți o aplicație mai intuitivă și mai ușor de folosit pentru ca știu cat de speriat și emoționat am fost în primele zile de studenție când încercam să urmăresc și sa înțeleg cum și unde are loc următoarea activitate.

Dar, nu în ultimul rând am ales această temă pentru a-mi dezvolta abilitățile de programare și experiența în dezvoltarea de aplicații front-end și de a înțelege cum se dezvoltă un proiect de calitate și durabil care să fie folosit de mii de utilizatori.

# I.Tehnologii utilizate

## I.1.VueJS

Pentru a realiza un site web sau o aplicație web, avem nevoie de HTML pentru componentele din pagină, CSS pentru design și JavaScript pentru funcționalitate.

VueJs este un framework open-source de JavaScript bazat pe principiul model-vedere-vederemodel folosit pentru a crea interfețe de utilizator și aplicații pe o singura pagina ( SPA – single- page aplication) și care folosește componente Vue pentru a extinde elementele HTML, pe care le încapsulează pentru a putea fi refolosite ori de cate ori este nevoie, oriunde în aplicație [1].

Componentele Vue pot fi create de dezvoltator, ele fiind compuse de un template reprezentant codul HTML al componentei , denumirea cu care va fi folosită și eventuale date și obiecte necesare.



Fig. I. 1 Exemplu componentă Vue

O dată creată, această componentă poate fi importată și folosită oriunde în proiect ca un tag HTML. În figura I.1 este prezentat un exemplu de componentă numită „button-counter” care înregistrează de cate ori a fost apăsat un buton. Dacă această componentă este integrată de mai multe ori, ea va retine propriul număr de apăsări, independentă de celelalte. în figura I.2 avem un astfel de exemplu.

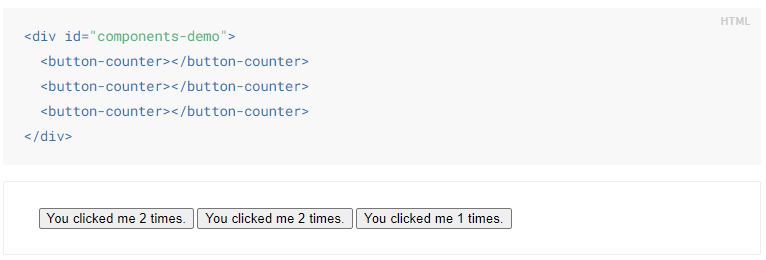


Fig. I. 2 Exemplu de reutilizare a componentelor Vue

Astfel, un proiect Vue este bazat pe create și refolosirea unor componente modulare, care reacționează la schimbări și folosește un DOM (Document Object Model ) virtual care modifică doar componentele sau părțile din pagină care trebuie actualizate, ceea ce reduce foarte mult din resursele necesare și face framework-ul sa fie unul foarte rapid și compact.

Vue are implementat un sistem de animații cu care se pot introduce numeroase efecte de tranziție la adăugarea, editarea sau eliminarea elementelor din DOM [1] și permite:

* Aplicarea automată de clase CSS pentru tranziții și animații
* Integrarea de librarii terță, cum ar fi Animate.css
* Folosirea limbajului Javascript pentru a manipula DOM-ul în timpul tranziției
* Integrarea de librării Javascript terță, cum ar fi Velocity.js

Pentru a avea folosi tranzițiile, codul căruia dorim sa îi aplicam tranziții trebuie încapsulat într-un tag numit *transition* și săalegem tipul de animații dorite. Un exemplu de implementare avem în figura I.3.



Fig. I. 3 Element cuprins într-un tag transition

Acum pentru a avea efectul dorit, trebuie să definim și codul CSS care să descrie animația care să se efectueze când paragraful cu textul *hello* va apare și dispare la apăsarea butonul de tip **t*oggle***. [3]



Fig. I. 4 Aplicarea unui efec de tranziție care durează 0.5 secunde

VueJS facilitează legătura dintre obiectele JavaScript și codul HTML folosindu-se de DOM-ul virtual și randare declarativă.

In figura I.5 se poate observa cum VueJS înlocuiește obiectul dintre acolade cu textul din variabila *message*, iar această, cum am menționat și mai sus, este reactivă: daca conținutul variabilei se modifică. VueJS va actualiza automat și textul afișat în browser, fără să fie necesară o actualizare explicită, dată de dezvoltator.

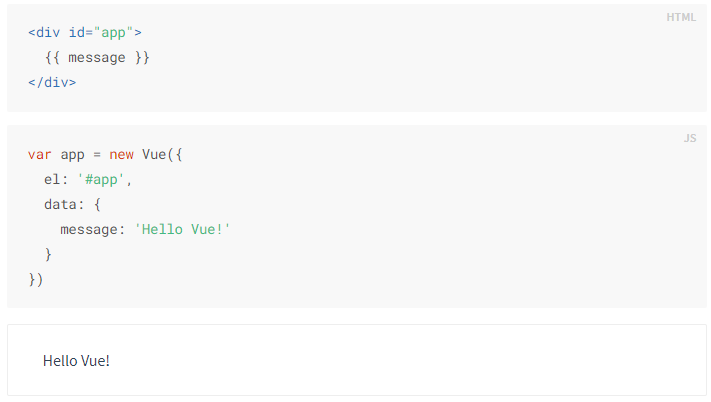


Fig. I. 5 Integrarea automată intre JavaScript și HTML oferită de VueJS

În figura I.3 se mai poate observa ***v-if=”show”***  care reprezintă o directivă VueJS pentru randare condiționată. ***Show*** este o denumirea unei variabile de tip Boolean, declarată în obiectul Vue, la fel cum este și ***message*** în figura I.5. Cand variabila ***show*** se modifică în **true** sau **false**, Vue afișează sau nu tag-ul HTML pe care este aplicat.

VueJS mai oferă și listare dinamică cu ajutorul directive ***v-for*** care repetă tag-ul HTML pe care este aplicat în functie de un vector, după cum se poate observa în figura I.6



Fig. I. 6 Exemplificarea directive v-for din VueJS

## I.2.TypeScript

TypeScript a apărut în 2012 ca și un superset sintactic pentru limbajul JavaScript și este menținut de către compania Microsoft, pentru a oferi suport opțional de tipuri de date. Codul TypeScript este transpilat în JavaScript de către **TypeScript Checker** sau cu ajutorul compilatorului Babel care poate fi invocat pentru conversie. [4]

VueJS este un framework care are ca suport nativ limbajul JavaScript, dar a primit suport pentru posibilitatea de a folosi Typescript și de creare a componentelor Vue sub formă de clase, de a declara variabile cu tipuri de date pentru a limita numarul posibilităților de a întâmpina probleme din cauza datelor, de a atribui interfețe obiectelor. [5]



Fig. I. 7 Componentă VueJS creată sub formă de folosind TypeScript

Deoarece JavaScript nu este un limbaj strongly typed, variabilele pot fi de orice tip, din neatenție sau din greșeală, acestea se pot modifica fără să ne dăm seama și să avem bug-uri sau comportament ciudat și să pierdem foarte mult timp încercând să descoperim sursa problemei, de aceea am ales să folosesc TypeScript, împreună cu VueJS, pentru a limita posibilitatea bug-urilor din cauza variabilelor pe care as putea să le asignez cu un al tip de date, lucru detectat de către compilator și raportat ca eroare.

Am mai ales TypeScript datorită obișnuinței de a lucra cu tipuri, dobândită în limbaje strongly typed precum C++, C# și Java, de pe parcursul celor 4 ani de studii universitare și pentru că, personal, consider că un proiect VueJS cu TypeScript este mai bine structurat și ușor de înțeles, datorită claselor, interfețelor și compilatorului care poate detecta unele erori înainte ca acestea să apară.



Fig. I. 8 Exemplu de interfată în TypeScript pentru VueJS

## I.3.Alte pachete și tehnologii

Prin natura sa, VueJS este un framework căruia i se pot adaugă module realizate de alți dezvoltatori sau parți terță. în cadrul proiectului am folosit următoarele module, mai importante:

* vue-class-component - folosit pentru TypeScript
* vue-loading-overlay - efect de încărcare pentru apeluri îndelungate
* vue-property-decorator - folosit pentru TypeScript
* vue-router - folosit pentru navigarea intre componente
* vuex - implementarea conceptului de Store pentru VueJS
* lodash - librarie de Javascript
* axios - client bazat pe promisiuni pentru apeluri HTTP
* font-awosome - librărie de fonturi și icon-uri
* boostrap - CSS framework folosit pentru un design flexibil

### I.3.1.BootstrapVue și Bootstrap

Pentru realizarea interfeței web, am folosit framework-ul open-source și gratuit de CSS, Bootstrap cu ajutorul BootstrapVue, o implementare în VueJS a librăriei Boostrap, pentru o integrare mai fluidă și mai rapidă.

Bootstrap foloseste un sistem flexbox grid împărtit pe 12 coloane cu ajutorul claselor **container**, **row** și **col**, cu ajutorul cărora se pot realiza interfețe care răspund schimbărilor de dimesiune a ecranului. [6]

Implementarea BootstrapVue foloseste componente VueJS sub formă de tag-uri HTML pentru utilizarea librăriei Bootstrap, despre deosebire de clase.

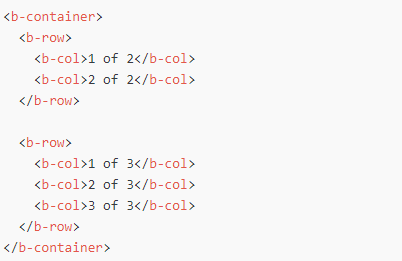


Fig. I. 9 Implementarea BootstrapVue a claselor în tag-uri

Am folosit Bootstrap pentru gruparea în tab-uri, tooltip-uri și bara de navigare care se extinde.

### I.3.2. HTML și SCSS (Sassy Cascading Style Sheets )

Deoarece HTML este sinonim cu Front-end și WebDevelopment, și nu cred să existe un site web fără acesta, trebuie sa spun câteva cuvinte și despre el.

HTML ( HyperText Markup Language ) este un limbaj de marcare, ajuns la versiunea 5.2 care este utilizat pentru realizarea de pagini și site-uri web ce vor fi interpretate/afișate cu ajutorul unui browser, pe aproape orice fel de dispozitiv, cu scopul de a prezenta informații, cum ar fi paragrafe, imagini, link-uri, tabele, etc. folosind cuvinte cheie intre semnele mai mic și mai mare cunoscute ca și **tag-uri HTML** [7].

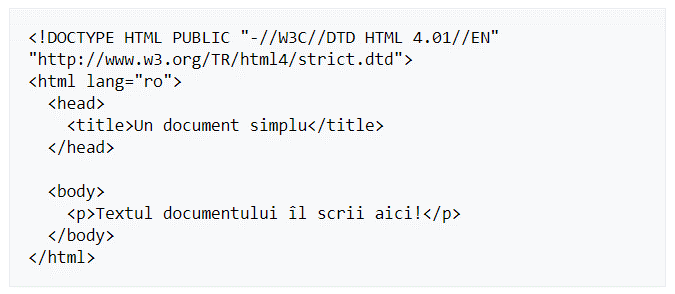


Fig. I. 10 Șablon document HTML

Pentru ca o pagină web să fie completă, avem nevoie și de un limbaj cu care să modifice elementele din documentul HTML și care sa aplice stiluri. Astfel avem nevoie și de SCSS, un super set al CSS ( Cascading Style Sheets ), pentru ca este foarte asemănător cu acesta din urmă, cu suport pentru variabile, și o ierarhie imbricată mult mai clară din punct de vedere vizual, cât și extindere/moștenire [8].



Fig. I. 11 Exemplu de cod SCSS și echivalentul său în CSS

# II.Proiectarea și implementarea

## II.1.Descriere generală

Proiectul a fost dezvoltat folosind Visual Studio Code și GitHub împreună cu Git pentru controlul versiunii, iar ca limbaj a fost folosit VueJS și TypeScript, SCSS și Bootstrap cu FontAwsome pentru icon-uri.

Aplicația preia datele legate de activități, cadre, săli de curs, facultăți și grupe de pe adresa ***orar.usv.ro/orar/vizualizare/date/xxx*** cu ajutorul pachetului Axios și le salvează folosind Vuex într-un spațiu de memorie temporar, numit Store, care poate fi accesat de oriunde din aplicație fără a fi nevoie de pasarea datelor între component într-o ierarhie părinte - copil.

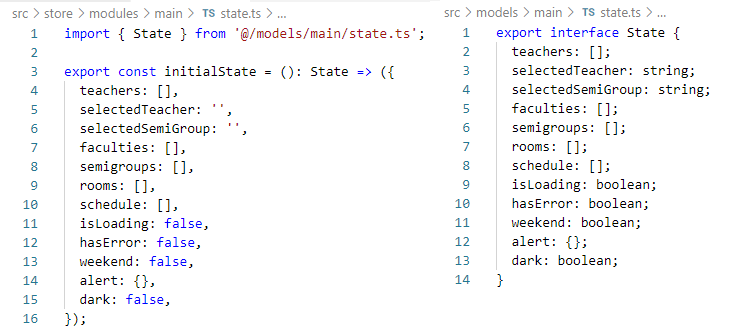


Fig. II. 1 - Interfața Store-ului și starea inițială

Preluarea datelor de la server se face prin următoarele apeluri:

**o*rar.usv.ro/orar/vizualizare/data/*** urmat de

* facultati.php - lista facultăților din Universitate
* subgrupe.php - lista completă de semi-grupe
* cadre.php - lista cadrelor didactice
* orarSPG.php?ID=XX&mod=YY - datele activităților după id și mod
  + mod **sala -** lista activităților dintr-o sală după id-ul ei
  + mod **prof -** lisa activităților unui cadru după id-ul său
  + mod **grupa -** lista activităților unei grupe după id-ul său

Datorită modului de apelare, datele vin împreună cu HEAD, meta și BODY care trebuie eliminate apoi împărțite în linii urmând să fie creat un vector de obiecte pe baza unor interfețe definite.

În figura II.2 se poate observa forma datelor primite pentru activitățile cadrului cu ID-ul 528, conf. dr. ing. Albu Angela, URL „**orar.usv.ro/orar/vizualizare/data/ orarSPG.php? ID=528&mod=prof**”**.**

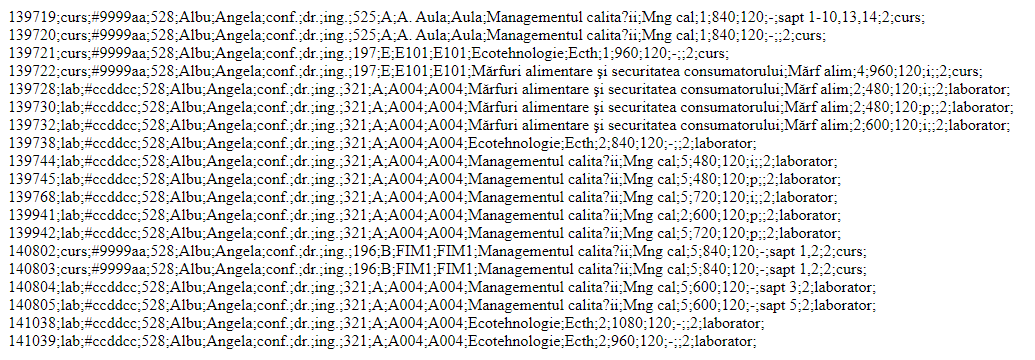


Fig. II. 2 - Exemplu de apel cadru didactic și răspunsul cu activitățile

După preluarea datelor și salvarea lor în Store Vuex, acestea sunt afișate într-un sub forma unui calendar pentru o săptămână, sau mai simplu spus, sub forma unui orar.

Vuex Store-ul este practic un container care stochează **starea** aplicației și se diferă de obiect global prin [9]:

* Vuex Store este reactiv. Când o componentă Vue accesează date din store, acesta se reacționă și își va modifica datele daca starea din store se modifică.
* Starea nu se poate modifica direct. Singurul mod în care starea store-ului se modifică este prin executarea unui mutații, care este înregistrată și poate fi urmărită pentru a înțelege mai bine aplicația.

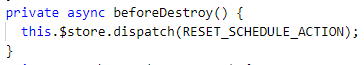


Fig. II. 3 - Executarea unei acțiuni care resetează conținutul store-ului

## II.2.Lista componentelor și structura acestora

Folosind componente VueJS am împărțit aplicația pe mai multe vederi prin care se poate naviga ca și pagini independente:

* Faculties - pagina cu lista facultăților și semi-grupelor
* Favorites - pagina orarelor favorite, alese de fiecare utilizator
* Main - prima pagina a aplicației cu legături către celelalte
* Rooms - pagina cu lista sălilor, precum laboratoare și amf.
* Schedule - pagina care afișează orarul grupelor sau profesorilor
* Teachers - pagina cu lista tuturor profesorilor din cadrul USV

Fiecare dintre aceste componente VueJS conțin una sau mai multe alte componente și scopul lor este pentru a fi folosite ca pagini individuale și pentru a facilita navigarea și accesarea lor prin link-uri URL.

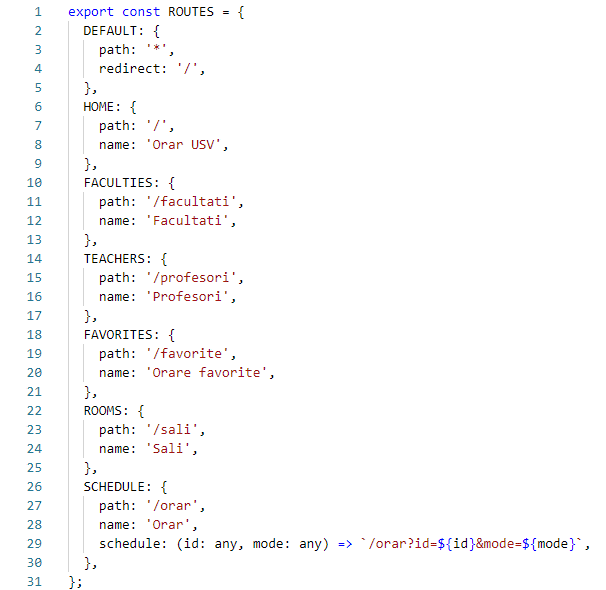


Fig. II. 4- Rutele în interiorul aplicației și calea acestora

Componenta Schedule este cea responsabilă pentru afișarea orarului, ea importă o componentă numite Course care este refolosită pentru afișarea fiecărei activități și care primește ca date de intrare, numite **props,** activitatea pe care să o afișeze.

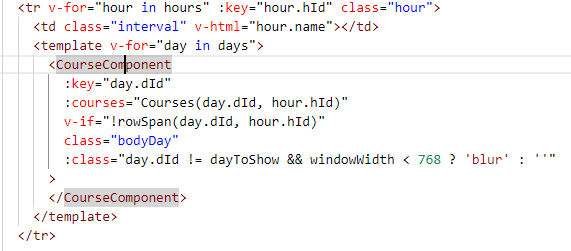


Fig. II. 5 Folosirea v-for și a unei componente Vue pentru afișarea informațiilor pentru activități

# III. Utilizarea aplicației

## III.1.Prima pagină

Aplicația vine cu un design nou și aerisit cu patru butoane mari de navigație între paginile site-ului.

Cele 4 butoane de navigație sunt:

* Favorite - orarele selectate de utilizator ca fiind favorite
* Facultăți - lista facultăților
* Profesori - lista profesorilor
* Săli - lista sălilor

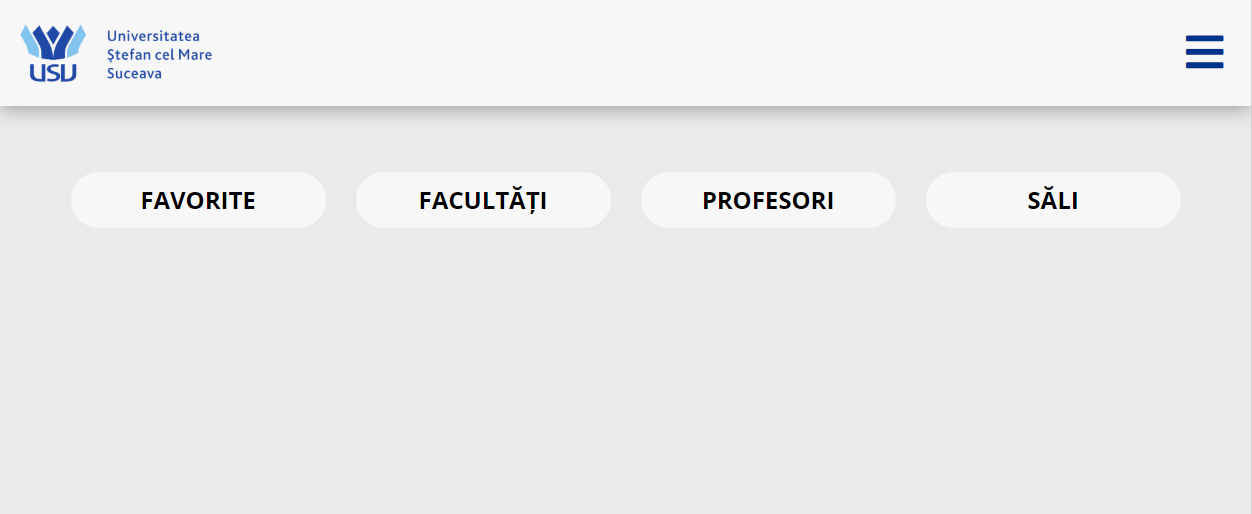


Fig. III. 1 - Prima pagină. varianta în modul Day

Bara de navigare conține următoarele opțiuni:

* Buton comutare între modul Day și modul Night
* Prima pagină
* Favorite
* Facultăți
* Profesori

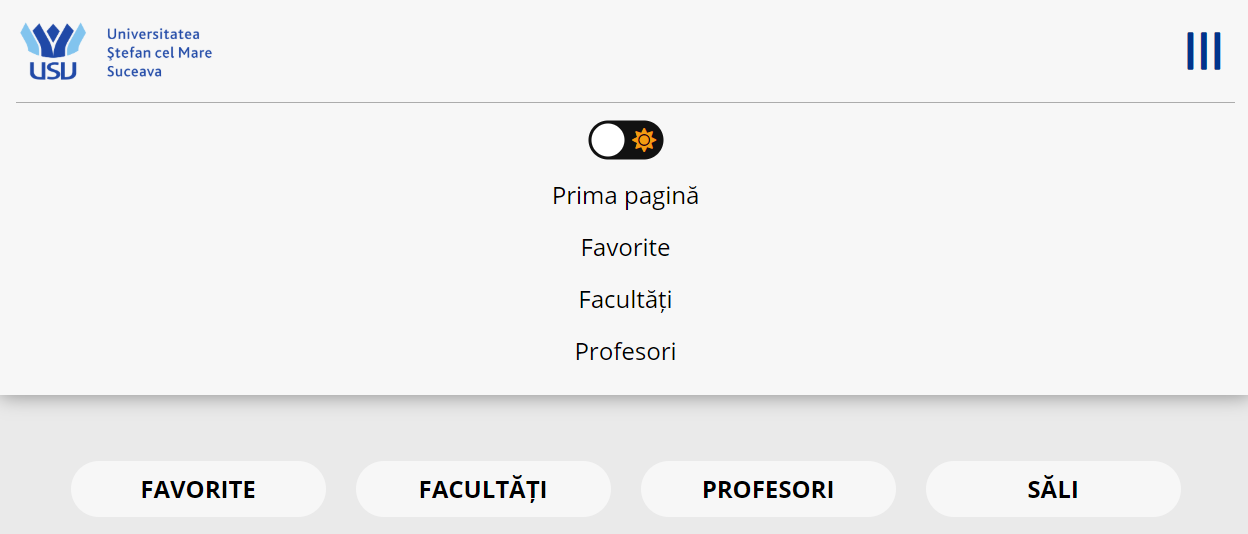


Fig. III. 2 - Bara de navigare deschisă, modul Day

În modul mobile, aplicația dimensionează butoanele și le poziționeză pe verticală pentru o mai bună vizibilitate.

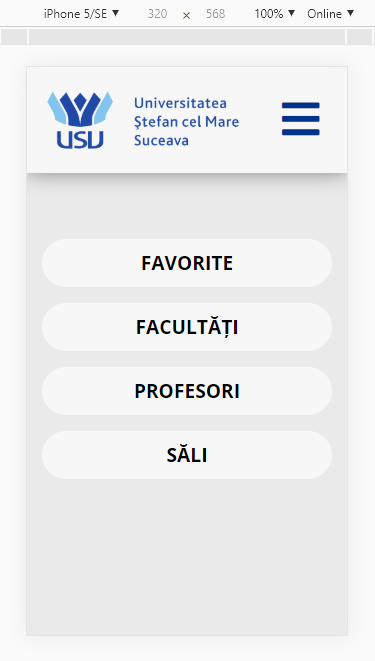


Fig. III. 3 - Vedere mobile pe un iPhone SE

## III.2.Modul Day/Night

În ultima perioadă, tot mai multe aplicații implementează, la cererea utilizatorilor un **mod Dark** sau **Night** pentru o interfață mai întunecată, ușoară pentru ochi și mai eficientă pe dispozitivele cu ecran AMOLED unde pixelii emit propria lumină, fiind mai eficient, mai ales din punct de vedere al energiei folosite de către dispozitiv. [10]

Deoarece, personal folosesc doar Dark Mode acolo unde este disponibil, am dorit să implementez și pentru Orar USV această funcționalitate activabilă și dezactivabilă de pe un buton ușor de accesat și care să se mențină între accesări, pe același dispozitiv fără a fi nevoie de autentificare, doar folosind LocalStorage pentru memorarea unei variabile de tip boolean și o interfață intuitivă cu un checkbox cu un design de tip **toggle** cu un soare și o lună.

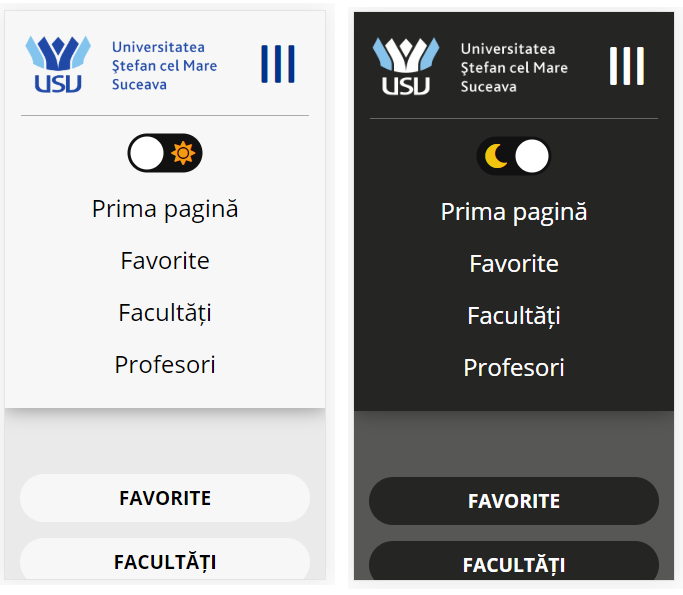


Fig. III. 4 - Exemplificare intre modurile Day - Night, vedere mobile

## III.3.Orare favorite

Din câte am observat, la colegii mei studenți, foarte multă lume își salvează o imagine screenshot a orarului și o folosesc pe acesta să urmărească activitățile. Acest lucru poate duce la probleme, mai ales în primele săptămâni, când structura orarului nu este definitivată și pot să apară schimbări, mai mici sau mai mari.

Astfel, una din soluțiile cu care am venit este de a da utilizatorilor posibilitatea să aleagă unul sau mai multe orare, cu grupe, profesor sau săli ca fiind favorite și să fie accesibile mult mai rapid, dintr-un meniu dedicat pentru a încuraja studenții să folosească aplicația și să fie tot timpul informați corect.

Lista orarelor favorite este salvată în LocalStorage și ramane până când utilizatorul alege sa golească istoricul site-urilor vizitate sau cookie-urile [11].



Fig. III. 5 - Listă orare favorite și listă goală

## III.4.Facultăți

În pagina **Facultăți**, este afișată o listă cu toate facultățile din USV, cu denumirea completă pe un dispozitiv cu ecran mai mare și denumirea scurtă pentru dispozitivele mobile.

La trecerea cursorului deasupra unui element din listă, acesta își mărește cu câteva procente dimensiunea și capătă un contur în culoarea clasica USV, albastră pentru un efect vizual plăcut.

La selectarea unei facultăți, aceasta dispare printr-un efect de fade-out spre stânga, iar lista grupelor aferente selecției apar fade-in spre dreapta.

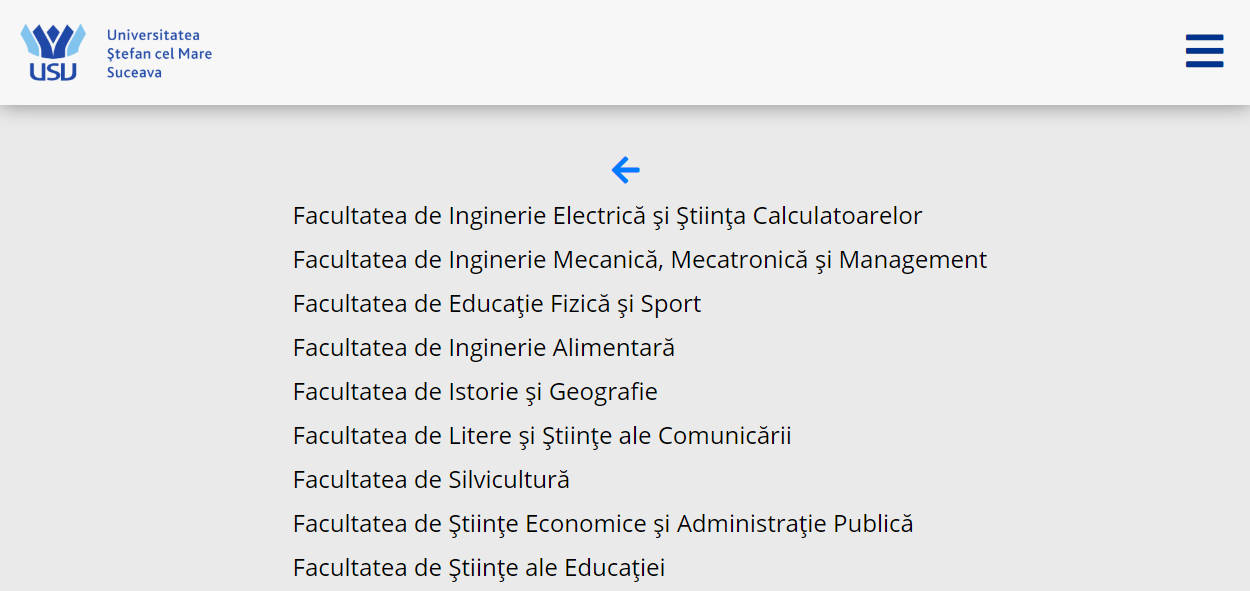


Fig. III. 6 - Lista facultăților disponibile

Lista grupelor este organizată pe tab-uri cu fiecare specializare, definite de denumirea scurtă a fiecărei specializări, ordonate de la licență la master și apoi studii postuniversitare, dacă este cazul,

Organizarea pe specializări este ordonată crescător după anul de studiu, pe bază de coloane care se restrâng în funcție de dimensiunea ecranului la o listă verticală și delimitată cu un spațiu după fiecare grupă.

Semi-grupele sunt centrate în interiorul div-ului și anul de studiu afișat deasupra. La fel ca și la lista anterioară, când cursorul este deasupra unei grupe, aceasta este mărită și conturată pentru a indica vizual această acțiune.

Atât pagina facultăților, cat și lista grupelor au cate un buton cu o săgeată intuitivă pentru a reveni la meniul anterior.

Selectarea unei grupe printr-un click sau touch, pe varianta mobile, va direcționa utilizatorul printr-un efect de tranziție fade-out, fade-in la pagina Schedule și va încărca activitățile aferente.

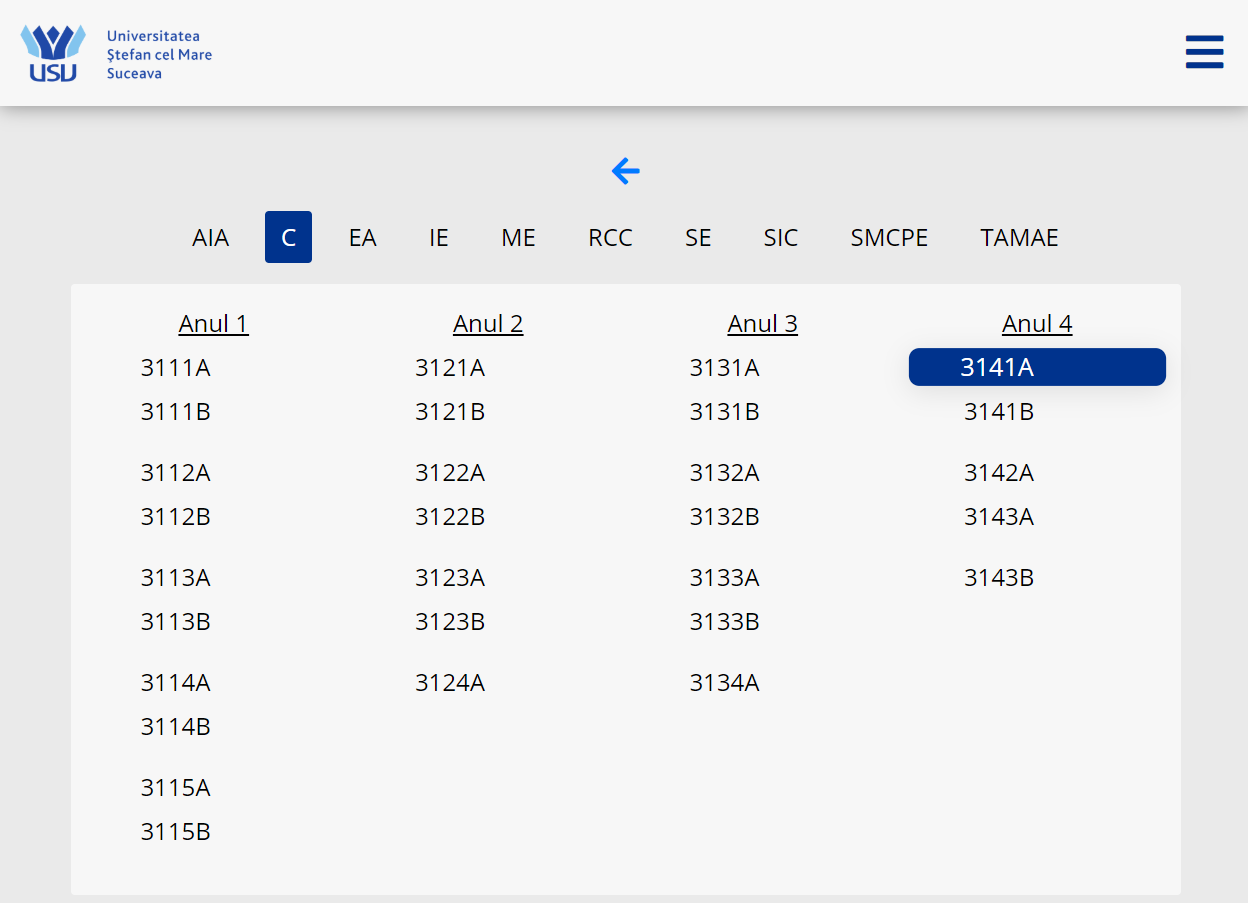


Fig. III. 7 - Lista specializărilor și grupele pentru Calculatoare, FIESC.

Astfel, alegerea grupei este mult mai ușoară și mult mai rapidă în comparație cu actualul model, care folosește un dropdown select, care poate fi foarte lung pentru unele facultăți, cum ar fi FIESC cu 19 semi-grupe de program Master și 67 semi-grupe de program Licență,

## III.5.Profesori / Cadre didactice

Lista profesorilor este acum acompaniată și de posibilitatea de a căuta un profesor după numele acestuia, pentru a ușura selecția, deoarece lista totală, la momentul redactării documentației este de 570 de cadre didactice pe toată Universitatea, cu același efect plăcut la trecea cursorului peste un cadru didactic din listă.

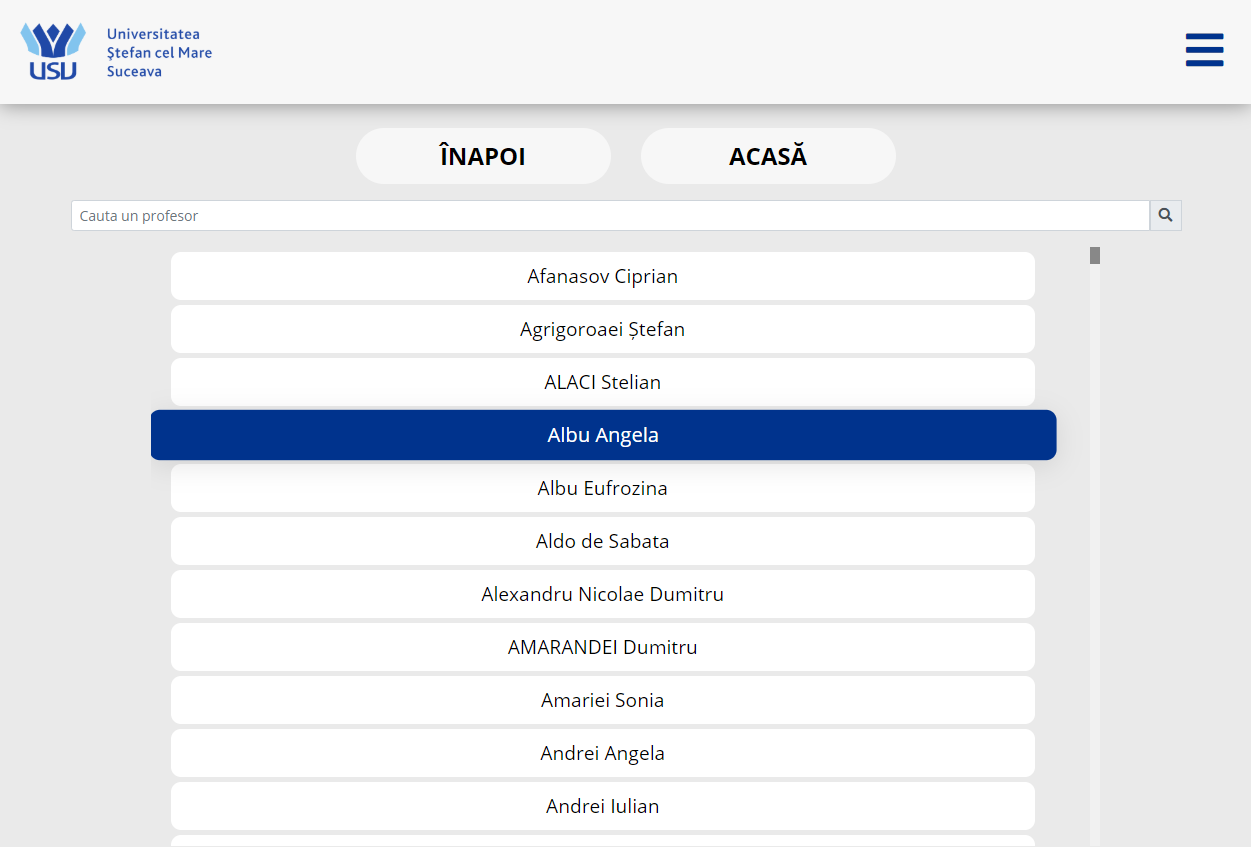


Fig. III. 8 - Lista celor 570 de cadre didactice.

Pentru varianta mobile, această listă păstrează toată funcționalitățile de pe versiunea desktop, dar într-un aspect mai restrâns.

În această secțiune se pot aduce foarte multe îmbunătățiri și funcții pentru a oferii utilizatorilor o experiență mai plăcută și o selecție mai rapidă a cadrului didactic pe care-l caută și poate mai multe efecte vizuale pentru a îmbunătății aspectul general al acestei pagini.

În figura III.9 se poate observa lista cadrelor didactice, potrivite pentru dimensiunea redusă a ecranelor, de exemplu, iPhone SE cu o rezoluție 320x568 și un raport de aspect 9:16.

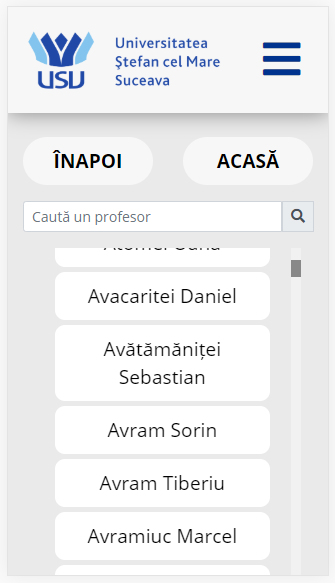


Fig. III. 9 - Lista celor 570 de cadre didactice, vedere mobile.

Selectarea unui cadru didactic printr-un click sau touch, pe varianta mobile, va direcționa utilizatorul printr-un efect de tranziție fade-out, fade-in la pagina Schedule și va încărca activitățile aferente cadrului selectat.

## III.6. Orarul

Partea cea mai importantă a acestei aplicației este afișarea activităților sub forma unui orar pe zilele săptămânii și design-ul acestuia să fie cat mai plăcut și usor de inteleg, în special pe partea de mobile, unde, cu sigurantă va fi folosit în cea mai mare parte a timpului de către studenți.

Design-ul pastrează aceeași paleta de culori pentru activități, acestea fiind împărțite in: **seminar**, **laborator**, **consultații**, **proiect** si, nu în ultimul rând, **curs**. îmbunătățind în special pe partea de mobile.

Orarul este poziționat central, cu un interval orar predefinit, de la 8:00 la 22:00, de luni pană vineri, și acolo unde este cazul, și în weekend, dacă cadrul, sala sau grupa are activități în zilele de sâmbătă sau duminică.

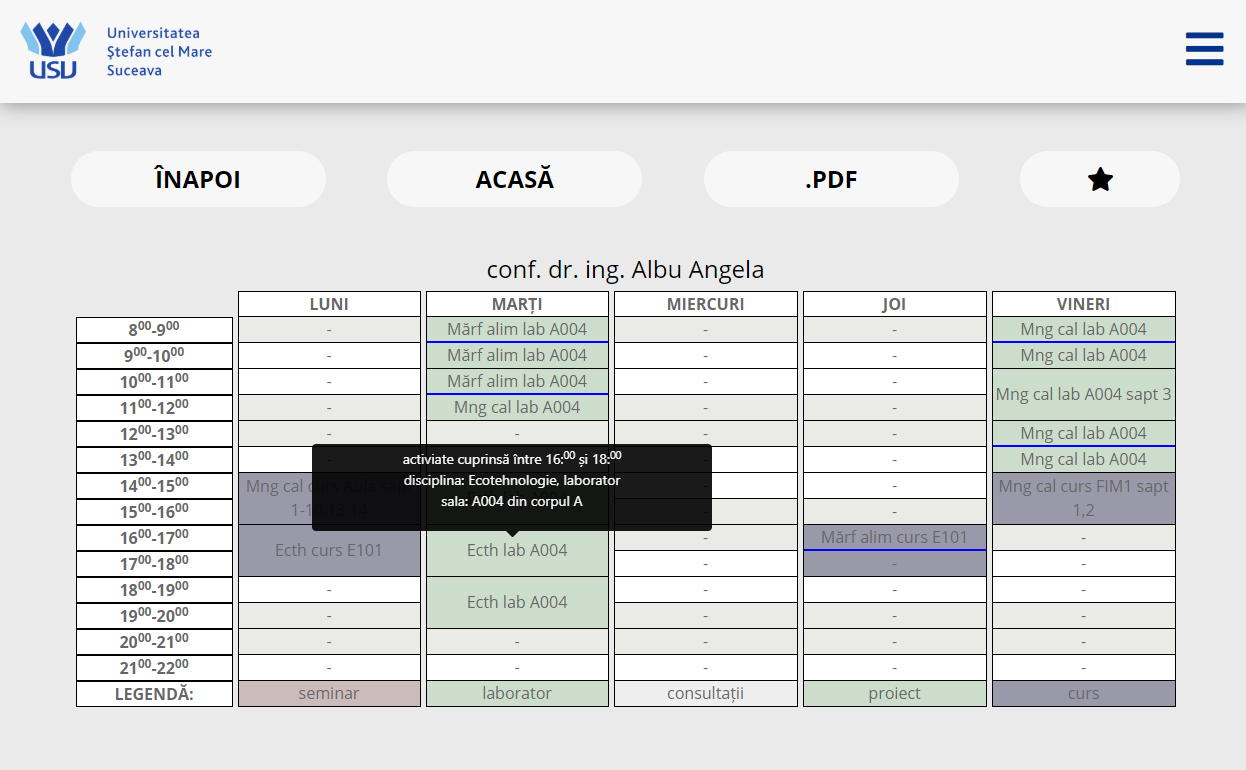


Fig. III. 10 - Activitățile unui cadru didactic și tooltip pentru informații suprmentare.

Pentru fiecare orar, indiferent de tipul acestuia, fie el de grupă, cadru didactic sau sală, are posibilitatea de a fi selectat ca și orar favorit printr-un click sau touch, pe partea de mobile, a steulei din colțul dreapta sus. Astfel, aceasta devine aurie, iar orarul va fi găsită în secțiunea **Favorite**, cum am prezentat anterior.

În cadrul variantei mobile, orarul este redus la o singură zi, iar navigarea între zile se face prin swipe, stânga sau dreapta, astfel modificându-se ziua curentă. În acest mod, orarul este mai usor de intels, nefiind nevoie de zoom-in sau zoom-out, acesta se potriveste automat dimensiunii ecranului utilizatorului.

Ziua de afișare este selectată automat ca fiind ziua curentă, iar în cazul weekend-ului, daca grupa, cadrul didactic, sau sala pe care o vizualizează utilizatorul nu are activitate, ziua de afișare va fi cea de luni.

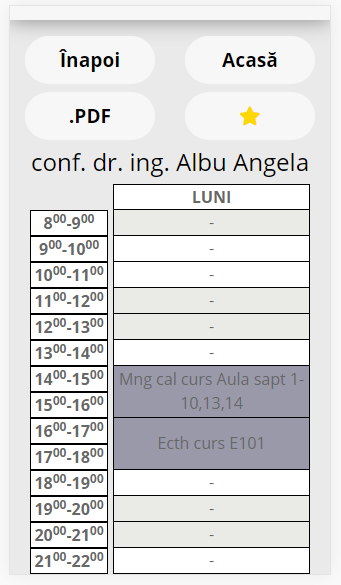


Fig. III. 11 - Orar favorit, versiunea mobile, pentru ziua de luni.

În figura III.11, orarul centrat ecranului, adaptându-se automat la o singură zi, iar acesta este setat ca și favorit, steluța fiind aurie.

# IV. Implementări ulterioare

Ca orice altă aplicație sau produs, proiectul poate fi îmbunătățit și dezvoltat în continuare, atât din punct de vedere vizual, cât si din punct de vedere al funcționalității, prin adăugarea de noi secțiuni, de opțiuni de personalizare sau de design.

Vă prezint o listă scurtă de implementări ulterioare

* Transformarea aplicației într-un PWA (Progresive Web Apps ) care sa poată fi „instalat” pe dispozitivele mobile și să se comporte ca o aplicație nativă.
* Actualizarea interfeței de administrare și crearea, editarea si ștergerea, într-un mod vizual, prin drag and drop a grupelor, cadrelor, sălilor fără să mai fie nevoie de interogări către baza de date scrise manual.
* Implementarea unei bot de IA ( inteligență artificială ) care sa genereze automat programările activităților în funcție de preferințele cadrelor didactice și de numărul grupelor pentru o repartizare cat mai corectă și mai bine organizată pentru a mulțumi atât cadre, cat si studenții.
* Crearea de orare personalizate de cate utilizatori, foarte folositor studenților cu credite restante care trebuie să refacă activitatea pentru materiile respective. Salvarea acestui orar personalizat să se efectueze fără a fi nevoie de un cont, doar printr-un cod unic general automat ( sau denumire unică ) și validat printr-o parolă introdusă la creare, și necesară doar in cazul editării acestuia.
* Optimizări pe partea vizuală pentru dispozitivele mobile, cât și funcțională.
* Solicitarea datelor de la back-end printr-un api și primirea acestora în format JSON ( JavaScript Object Notation ) și înlăturarea părților de PHP.
* Adăugarea mai multor opțiuni de navigare, cum ar fi posibilitatea trecerii de la un orar la altul prin selectarea unei profesor sau sală și invers.
* Implementarea orarului pe specializare, în care să fie prezentate toate grupele din respectiva specializare.
* Posibilitatea utilizatorilor de a seta anumite preferințe pentru aplicație, precum:
  + Alegerea paginii de start, între prima pagina, cadre didactice, săli, favorite sau chiar un anumite orar.
  + Posibilitatea de a dezactiva auto-selectarea zilei curente ca zi de start pentru versiunea mobile a aplicației.
  + Abilitatea de a atribui o culoare preferențială pentru oricare element din aplicație.
  + Import / Export al preferințelor într-un fișier CSV sau prin cod unic generat la export.
* Buton de feedback / raportează o problemă pentru ca utilizatorii să poate comunica direct cu dezvoltatorul în cazul unor probleme sau păreri ce doresc a fi împărtășite.
* Îmbunătățire suport browsere mai vechi, precum IE8, IE9 și IE11 cât și pentru dispozitivele mobile cu dimensiuni foarte mici sau foarte mari.

Lista de funcții suplimentare poate fi mărită in funcție de feedback-ul și cerințele studenților și cadrelor didactice, după ce această aplicație va fi dată în folosință.

# Concluzii

Prin această aplicație se ușurează folosirea, cat și înțelegerea Orarului USV, printr-o interfață nouă și aerisită, cu funcționalități noi, design plăcut, adaptabil rezoluției ecranului de pe care este accesat.

Framework-ul ales va permite adăugarea cu ușurință a unor funcționalități noi, cat și rezolvarea rapide de bug-uri, în cazul în care acestea apar.

Din punctul meu de vedere, diferențele între aplicația actuală și implementarea mea sunt foarte mari și bine venite, și aplicația va avea un real succes, mai ales in rândul bobocilor.

Din perspectiva mea, această aplicație m-au ajutat să îmi dezvolt cunoștințele despre cum se dezvoltă și cum se adaptează un proiect de dimensiuni mari De asemenea, mă voi putea baza pe aceste cunoștințe dobândite, și în realizarea proiectelor personale și profesionale pe care le voi realiza în viitor.

# Bibliografie

|  |  |
| --- | --- |
| [1] | VueJS, „Vue.JS,” 2020. [Interactiv]. Available: https://en.wikipedia.org/wiki/Vue.js. |
| [2] | Stack Exchange, Inc., "Stack Overflow Trends," 07 2020. [Online]. Available: https://insights.stackoverflow.com/trends?tags=vue.js%2Cangular%2Creactjs%2Ctypescript. |
| [3] | G. VueJS, „Transitions & Animation,” 2020. [Interactiv]. Available: https://vuejs.org/v2/guide/transitions.html. |
| [4] | TypeScript, „TypeScript,” 2020. [Interactiv]. Available: https://ro.wikipedia.org/wiki/TypeScript. |
| [5] | P. HS, „How to write a Vue.js app completely in TypeScript,” 20 01 2020. [Interactiv]. Available: https://blog.logrocket.com/how-to-write-a-vue-js-app-completely-in-typescript/. |
| [6] | BootstrapVue, „BoostrapVue,” 2020. [Interactiv]. Available: https://bootstrap-vue.org/. |
| [7] | W3School, „HTML Introduction,” 2020. [Interactiv]. Available: https://www.w3schools.com/html/html\_intro.asp. |
| [8] | Sass-Lang, „Sass: Sass Basic,” 2020. [Interactiv]. Available: https://sass-lang.com/guide. |
| [9] | G. Vuex, „Getting Started | VUex,” 2020. [Interactiv]. Available: https://vuex.vuejs.org/guide/. |
| [10] | T. Lyles, „Google's app is getting a dark mode,” 19 03 2020. [Interactiv]. Available: https://www.theverge.com/2020/5/19/21264056/google-app-dark-mode-android-ios. |
| [11] | Javascript.info, „LocalStorage, sessionStorage,” The Modern JavaScript Tutorial, 2020. [Interactiv]. Available: https://javascript.info/localstorage#sessionstorage. |