**Documantatie**

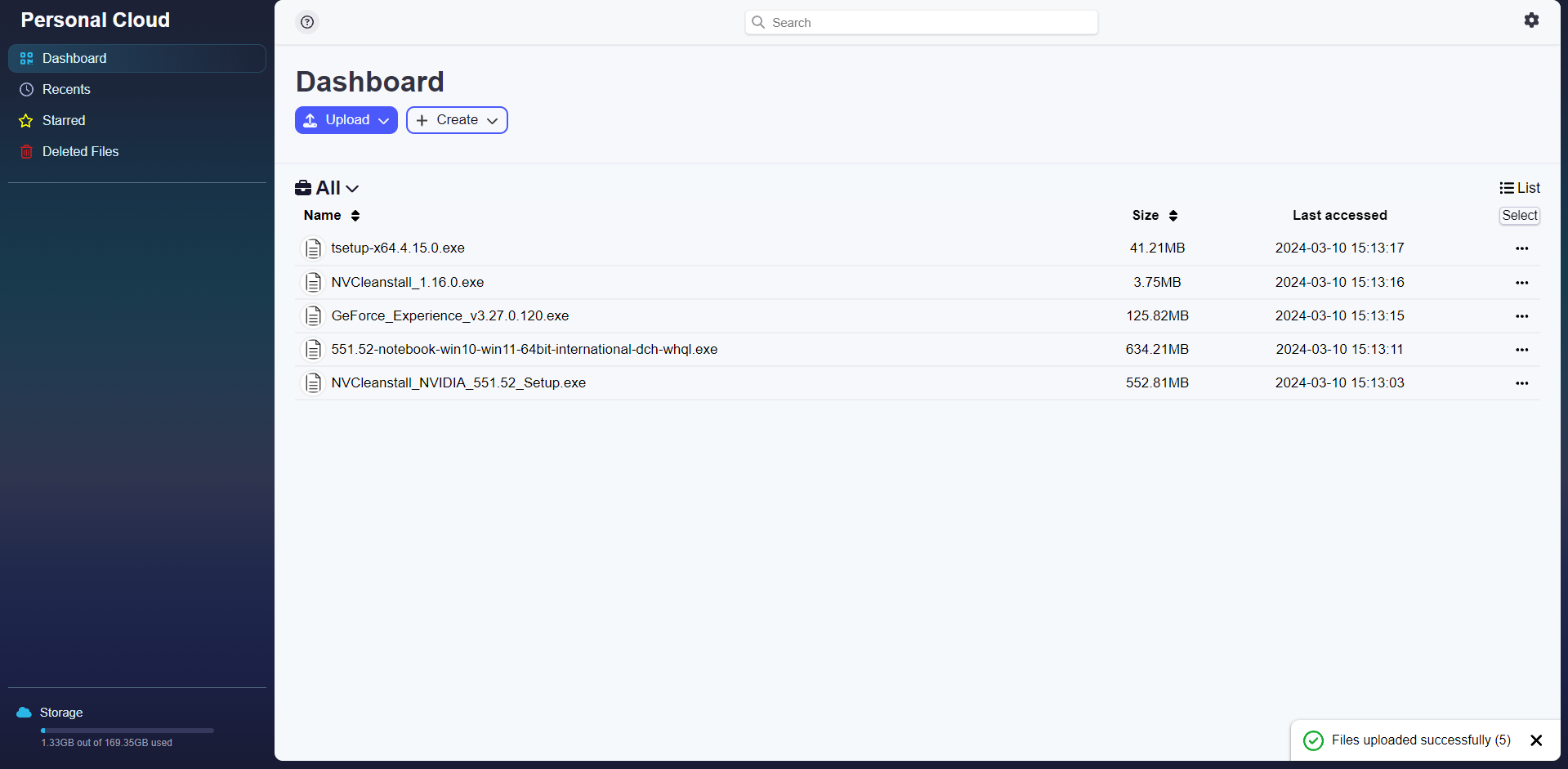
**1.SPECIFICAȚII GENERALE**

Proiectul "Cloud Storage" este o aplicație web concepută pentru stocarea de fișiere sau dosare personale de orice dimensiune.

Accesarea aplicației web necesită deținerea unui cont pentru a putea accesa fișierele personale. În partea stângă a paginii se găsesc butoanele de navigare: "Dashboard" (aici se găsesc toate fișierele și dosarele, cu excepția celor șterse), "Starred" (unde se găsesc fișierele și dosarele adăugate la favorite), "Recents" (pagină dedicată fișierelor accesate în ultimele 7 zile) și "Deleted" (conține fișierele și dosarele care au fost șterse în ultimele 7 zile). De pe pagina "Dashboard", utilizatorul are posibilitatea să încarce un dosar sau un fișier sau să creeze un dosar gol, căruia trebuie să-i atribuie un nume și să selecteze locația acestuia. De pe orice altă pagină, fișierele/dosarele pot fi afișate pe categorii (fișiere, dosare sau ambele). De asemenea, acestea pot fi sortate crescător sau descrescător în funcție de nume, mărime sau în funcție de ultima accesare, iar sortarea implicită este în funcție de ora încărcării fișierului/dosarului. De asemenea, fișierele/dosarele pot fi vizualizate sub formă de tabel sau listă.

Fiecare fișier/dosar are o listă de opțiuni care poate fi afișată făcând clic pe cele trei puncte din dreapta acestuia. Opțiunile pentru un fișier/dosar constau în: deschiderea dosarului (care poate fi deschis și prin dublu click), redenumire, schimbarea locației acestuia, adăugarea/ștergerea de la favorite, descărcare, generarea/ștergerea unui link temporar de descărcare valabil 60 de minute, recuperare (în cazul în care fișierul/dosarul este șters) si ștergere.

Fișierele și documentele pot fi identificate mai ușor folosind bara de navigație, oferind utilizatorului posibilitatea de a căuta într-un dosar, pe o pagină sau prin toate dosarele și fișierele.



**2.LIMBAJE SI TEHNOLOGII FOLOSITE**

Vue js - un framework JavaScript progresiv pentru construirea de interfete de utilizator si aplicatii sofisticate pe o singura pagina, usor de invatat si de integrat cu alte biblioteci de functii JavaScript sau proiecte existente. Se axeaza in principal pe partea de vizualizare (front-end, client side).

HTML - un limbaj de marcăre utilizat pentru structurarea și afișarea conținutului pe paginile web.

CSS - un limbaj de stilizare a elementelor si tagurilor html.

Node js - mediu de executie JavaScript inafara browserului care este folosit pe partea de server impreuna cu libraria Express, o librarie pentru Node js folosit pentru a construi RESTfull API’s. Aceasta tehnologie ofera posibilitatea de a folosi JavaScript atat pentru front-end cat si pentru back-end.

MySQL - limbaj standardizat utilizat pentru gestionarea bazelor de date relaționale. SQL oferă comenzi și funcții pentru crearea, modificarea și interogarea bazelor de date, precum și pentru gestionarea drepturilor de acces și a securității datelor.

**2.ARHITECTURA APLICATIEI**

**A.** Arhitectura generala a aplicatiei

Aplicația web Cloud Storage este structurată în două mari părți: **frontend** și **backend**.

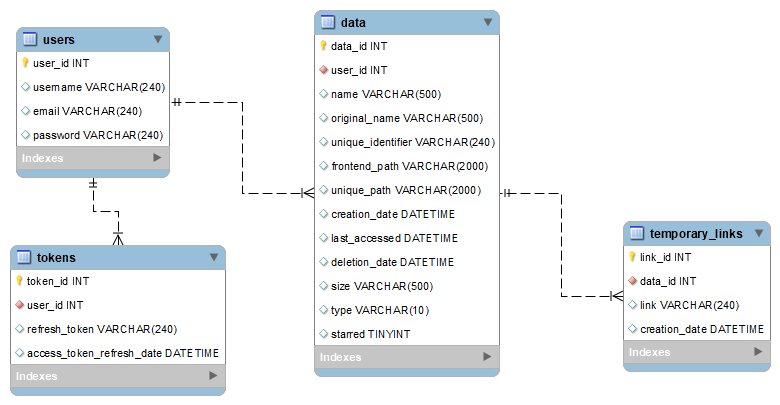


**Frontend**, sau partea client, este dezvoltată în Vue js și este responsabilă de interfața cu utilizatorul, de tot ce acesta poate vedea și elementele cu care acesta poate să interacționeze. De asemenea, este responsabilă si de logica aplicației care rulează în browserul utilizatorului (validarea datelor de intrare, manipularea evenimentelor, actualizarea interfeței etc.). În acest scop se folosesc diferite componente cu ajutorul cărora se controlează fluxul de date în aplicație. Pentru a comunica cu backend-ul în scopul transferului de date, se folosește biblioteca Axios, ce permite efectuarea cererilor de tip REST Api.

**Backend**, sau partea server, dezvoltat cu Node js, se ocupă cu gestionarea, prelucrarea și stocarea datelor, de logica și de securitatea acesteia. Partea server expune patru controlere ce conțin punctele finale către care Axios trimite cererile Http, având rolul de a prelua și de a răspunde acestor cereri. De asemenea, pe partea de backend se realizeaza si conexiunea cu baza de date MySQL si poate sa modifice, sa stearga sau sa adauge date noi in baza de date.

**B.** Structura bazei de date

Baza de date contine 4 tabele: tokens, temporary\_links, users si data. Tabelul users si tabelul data sunt relationate printr-o relatie one-to-many.



Tabelul "tokens": Acesta token-urile folosite pentru autentificarea utilizatorilor. Aceste token-uri sunt esențiale pentru gestionarea și autentificarea sesiunilor utilizatorilor.

Tabelul "temporary\_links": Acesta conține link-uri generate temporar, care sunt programate să fie șterse automat după 60 de minute. Acest tabel este folosit pentru a memora link-urile generate pentru descarcarea unui fisier/dosar.

Tabelul "users": Acest tabel stochează informațiile referitoare la utilizatori, cum ar fi numele de utilizator, adresa de email și parola. Relația "one-to-many" dintre tabelul "users" și "data" sugerează că un utilizator poate avea mai multe înregistrări asociate în tabelul "data".

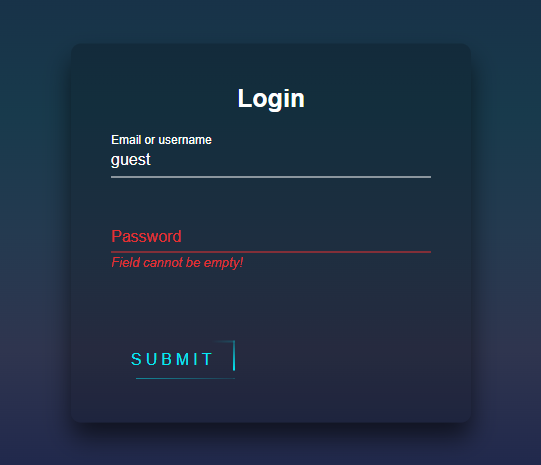
Tabelul "data": Acesta conține informații despre locația fisierelor și dosarelor pe server și pe partea de client. De asemenea, stochează informații despre dacă fișierul/dosarul este adăugat la favorite și dacă se încadrează în fișierele/dosarele accesate recent. Relaționarea cu tabelul "users" indică că fiecare înregistrare din "data" este asociată cu un utilizator specific.

**3.DESCRIEREA FUNCTIONALITATIILOR APLICATIEI**

**A.** Autentificarea

Procesul de autentificare a unui utilizator se realizează folosind JSON Web Token (JWT), un șir securizat semnat cu o cheie secretă. Acest șir conține id-ul utilizatorului. După validarea numelui de utilizator/emailului și a parolei (criptată cu biblioteca argon2), se creează dosarele necesare pentru stocarea fișierelor încărcate de către utilizator, în cazul în care acestea nu există deja. Totodată, se generează un token de acces (memorat într-un cookie) și unul suplimentar pentru actualizarea token-ului de acces.

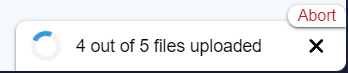
Token-ul de acces este valid pentru 30 de minute, după care, cu ajutorul token-ului de refresh, se generează unul nou. Această abordare are rolul de a limita accesul unui potențial atacator care a obținut token-ul de acces, oferindu-i acces doar pentru maxim 30 de minute. De asemenea, dacă un utilizator își închide sesiunea sau token-ul de refresh este șters din baza de date (de exemplu, în cazul deconectării), accesul va fi blocat.



**B.** Incarcare fisierelor si a dosarelor

Procesul de încărcare și gestionare a fișierelor și dosarelor în aplicație implică utilizarea a două input-uri distincte, unul dedicat fișierelor și celălalt dosarelor. Diferența esențială între cele două constă în faptul că primul input permite selectarea fișierelor, în timp ce cel de-al doilea permite selectarea dosarelor. Ambele input-uri conțin fișierele selectate, cu excepția că fișierele din input-ul pentru dosare au o proprietate suplimentară care indică locația fiecărui fișier în dosarul respectiv. Această proprietate este ulterior utilizată pentru a crea dosarele.

Pentru trimiterea fișierelor, se folosește biblioteca Resumable.js, care permite transferul de fișiere în bucăți folosind protocolul HTTP. În cazul în care încărcarea unei bucăți eșuează, procesul este reluat până când încărcarea este finalizată. Această abordare asigură reluarea automată a încărcării în caz de pierdere a conexiunii la rețea. De asemenea, se poate vedea si cate fisiere su fost incarcate.Biblioteca oferă, de asemenea, informații despre progresul încărcării și le oferă utilizatorilor opțiunea de a opri procesul de încărcare.



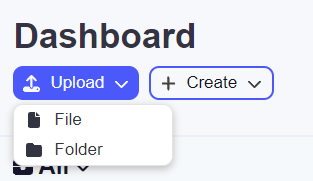
Dupa incarcare tuturor fisierelor apare urmatorul mesaj:



Pe partea serverului, se utilizează biblioteca multer pentru stocarea bucăților de fișiere pe disk, într-un folder numit "chunks". Se adaugă un obiect într-un array care reprezintă datele despre un fișier, inclusiv numele original, un nume unic generat la momentul respectiv, locația acestuia (o locație unică și una originală), numărul de părți pe care trebuie să le primească, un array cu denumirea părților memorate pe disk și dimensiunea acestuia. După primirea tuturor bucăților, acestea sunt asamblate într-un alt folder numit "files", una câte una, pentru a utiliza cât mai puțină memorie și a îmbunătăți performanța. După asamblare, bucățile sunt șterse, iar datele despre fișier sunt memorate în baza de date.

În cazul în care utilizatorul încarcă un dosar, fișierele vor avea o proprietate nouă care indică locația acestora în dosar (de exemplu, "documente/certificate"). Aceasta este folosită pentru a crea un obiect care conține datele despre dosar, în cinci etape distincte:

1. Se utilizează metoda split pentru a genera un array cu dosare din proprietatea fisierului. Array-ul este parcurs, și pentru fiecare dosar se creează un obiect ce reprezintă datele despre acel dosar, cu proprietăți precum numele original, un nume unic generat la moment, locația formată din denumirile originale ale dosarelor, locația formată din denumirile unice ale documentelor (generate ulterior), dimensiunea (extrasă mai târziu), utilizatorul căruia îi aparține și poziția dosarului.
2. Se parcurge array-ul cu dosare și se creează locația unică a acestora. Se folosesc două structuri repetitive de tip for pentru a verifica fiecare dosar cu toate celelalte și se verifică dacă folderul respectiv se află în celălalt. Locația unică este creată în acest proces.
3. Se setează locația unică a fișierelor în funcție de locația unică a dosarelor. Dacă un fișier se află în dosarul cu numele x, atunci locația unica a fișierului este locația unica a dosarului x + denumirea unica a acestuia.
4. Se memorează dosarele în baza de date.
5. După fiecare fișier memorat, se adaugă dimensiunea acestuia la dimensiunea dosarelor în care se află.

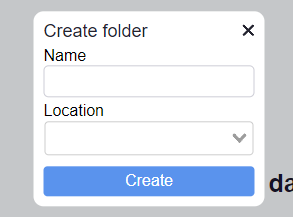


**C.** Crearea unui dosar

Procesul de creare a unui dosar se desfășoară prin intermediul unui modal. La deschiderea acestui modal, este apelată o funcție care, utilizând biblioteca Axios, extrage toate dosarele existente. Modalul cuprinde un câmp în care utilizatorul introduce numele dosarului și un selector pentru a alege locația dosarului. La apăsarea butonului "Creează", se inițiază o cerere către server.

Pe partea serverului, se generează un nume unic pentru noul dosar, iar locația este setată conform selecției utilizatorului. Aceste informații sunt incluse în cererea către server. Procesul de generare a numelor unice poate implica utilizarea unor metode specifice, cum ar fi utilizarea unui identificator unic universal (UUID).

În final, serverul primește cererea, validează și prelucrează informațiile, creând noul dosar cu numele unic și locația specificate.



**C.** Primirea dosarelor si a fisierelor

La accesarea unei pagini, este apelată o funcție care returnează un JSON conținând toate informațiile despre fișiere și dosare, precum și date despre memorie (memoria totală și cea folosită). Funcția primește un parametru care indică denumirea paginii pentru care trebuie extrase datele. De exemplu, pentru pagina "dashboard", se extrag toate fișierele și dosarele, cu excepția celor din coșul de gunoi. Pentru pagina "starred", se extrag fișierele și dosarele care au proprietatea "starred" setată la 1, și așa mai departe.

Acest parametru este trimis către server, unde se efectuează o interogare la baza de date pentru a obține toate fișierele și dosarele deținute de utilizator. Acestea sunt parcurse și categorizate, iar în același timp se calculează memoria ocupată de acele fișiere. Ulterior, se elimină toate fișierele și dosarele care se află într-un alt dosar încadrat în aceeași categorie. Informațiile procesate sunt apoi trimise înapoi la client.

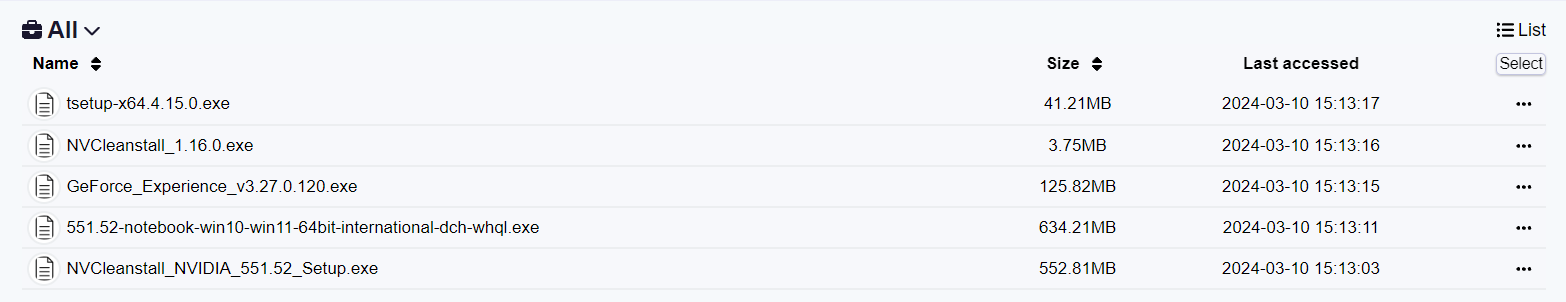
Datele despre memorie sunt memorate în Vuex Store, o bibliotecă de stare globală în Vue.js, care permite crearea variabilelor globale și accesul la acestea din orice parte a aplicației. JSON-ul conținând informații despre fișiere și dosare este parcurs și afișat pe ecran pentru a furniza utilizatorului o prezentare coerentă și ușor de înțeles a conținutului asociat paginii accesate.

**D.** Vizualizarea fisierelor si a dosarelor

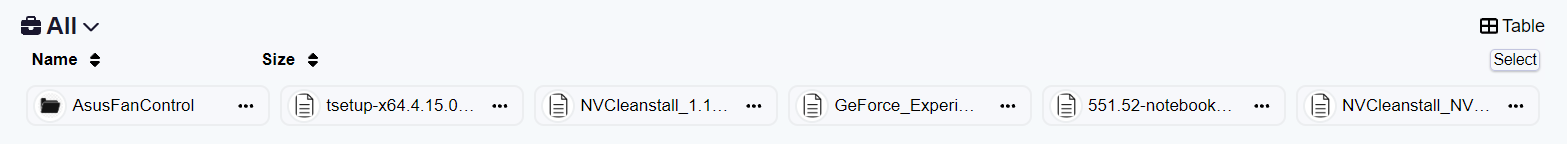
Pentru categorizarea datelor afișate, utilizatorul poate opta să vizualizeze doar fișiere, doar dosare sau toate elementele disponibile. Această categorizare este implementată prin utilizarea funcției filter(), care returnează un nou array conținând toate elementele care îndeplinesc condiția impusă.

De asemenea, utilizatorul are opțiunea de a alege între două moduri de vizualizare: sub forma unei tabele sau sub forma unei liste. Această funcționalitate este gestionată printr-o variabilă de tip boolean, care indică dacă trebuie să fie afișată o vizualizare de tip tabel sau de tip listă. Această decizie este implementată în interfața utilizatorului utilizând directivele v-if și v-else.

Format lista:



Format table:



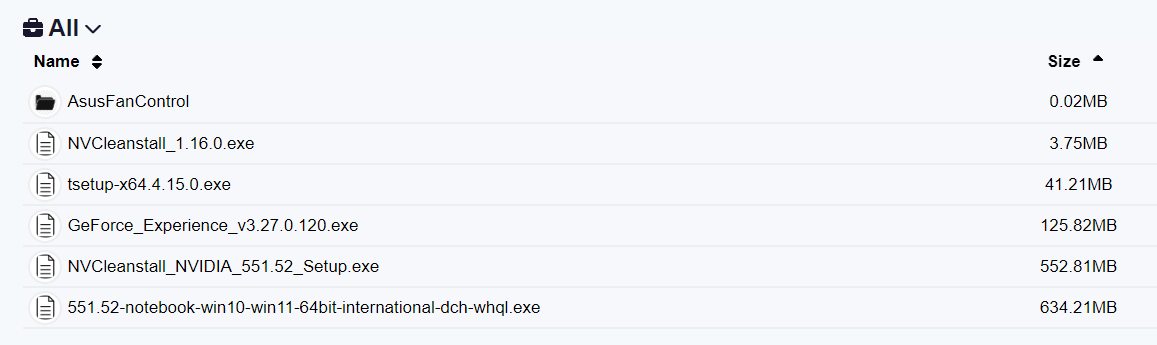
**E.** Sortarea fisierelor si a dosarelor

Fișierele și dosarele sunt sortate în mod implicit după data de încărcare, folosind metoda sort() pe obiectul JSON și getTime() pe câmpul care conține data încărcării pentru a realiza o sortare precisă.

De asemenea, utilizatorul are opțiunea de a realiza sortări suplimentare în funcție de nume și dimensiune. Aceste sortări pot fi efectuate în ordine crescătoare sau descrescătoare folosind funcția sort(). Sortarea după nume se realizează prin compararea șirurilor de caractere, iar sortarea după dimensiune se bazează pe comparația valorilor numerice ale timpului.

În plus, datele afișate pot fi categorizate în funcție de tipul lor. Utilizatorul are posibilitatea să aleagă să vizualizeze doar fișiere, doar dosare sau toate elementele disponibile.

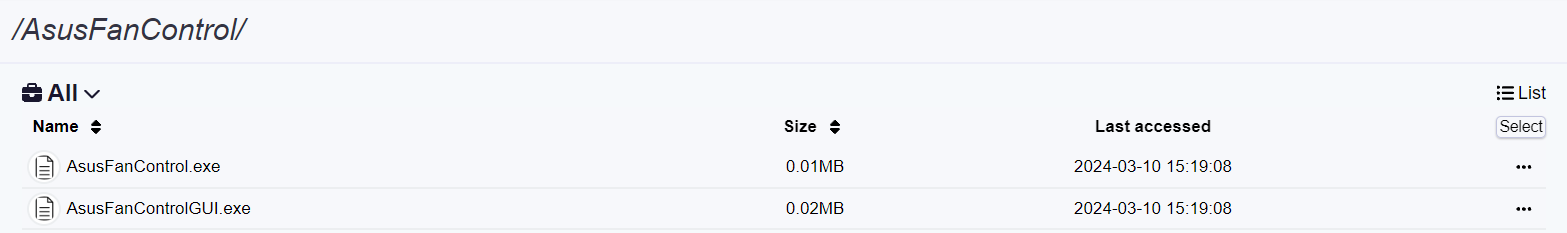
Un exemplu in care fisierele/dosarele sunt ordonate crescator in functie de marime.



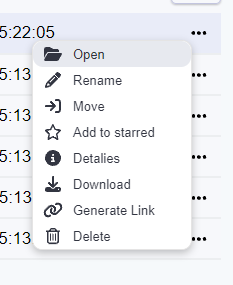
**F.** Primirea dosarelor si a fisierelor dintr-un dosar

Prin efectuarea unui dublu clic pe un dosar sau deschiderea meniului cu opțiuni și selectarea opțiunii "Open", utilizatorul va fi redirecționat către o pagină nouă, iar router-ul va primi ca parametru numele unic al dosarului. La accesarea acestei pagini, un apel către server este efectuat, transmitând un request cu numele dosarului către baza de date.

Server-ul procesează cererea prin intermediul unei interogări a bazei de date, extrăgând toate fișierele și dosarele care au ca locație unică numele dosarului specificat. Aceste informații sunt returnate sub forma unui obiect JSON și sunt apoi afișate pe pagina respective.



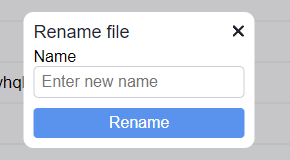
**G.** Folosirea optiunilor fisierelor si a dosarelor

Când un utilizator face clic pe butonul care deschide lista de opțiuni, există o variabilă care stochează datele referitoare la fișierul sau dosarul pentru care a fost deschisă lista de opțiuni. În momentul selectării unei opțiuni, se inițiază una dintre cele două acțiuni disponibile: directă sau indirectă.

Acțiunea directă presupune apelarea unei funcții și efectuarea unei cereri la server imediat. În schimb, acțiunea indirectă implică deschiderea unui modal în care utilizatorul trebuie să introducă anumite date, iar doar după apăsarea butonului din partea de jos se efectuează cererea la server.

**H.** Redenumirea unui fisier/dosar

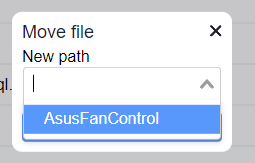
Procesul de redenumire a unui fișier sau dosar începe prin deschiderea unui modal care conține un câmp de introducere text, unde utilizatorul poate introduce noul nume al fișierului sau dosarului. În cazul în care utilizatorul dorește să redenumească un fișier, extensia acestuia va fi păstrată automat. La apăsarea butonului "Rename", se inițiază o cerere către server, în cadrul căreia sunt transmise noul nume și denumirea unică a fișierului sau dosarului, iar server-ul actualizeaza baza de date.



**I.** Schimbarea locatiei unui fisier/dosar

Procesul de mutare a unui fișier sau dosar implică deschiderea unui modal care conține un buton și un câmp de tip selector provenit din biblioteca Vue-Selector. Acest câmp permite utilizatorului să selecteze dosarul în care dorește să plaseze fișierul sau dosarul respectiv. La apăsarea butonului "Move", se inițiază o cerere către server pentru a trimite dosarul sau fișierul către destinația dorită.

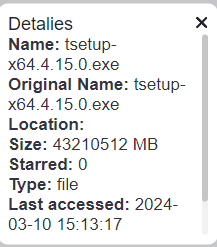
În backend, se efectuează o interogare la baza de date pentru a extrage toate fișierele și dosarele asociate utilizatorului. Acestea sunt parcurse, iar în cazul în care utilizatorul dorește să schimbe locația unui dosar, se modifică și locația fișierelor și dosarelor din acel dosar. Prin verificarea locației unice a unui fișier sau dosar, se determină dosarul în care se află. Dacă condiția este îndeplinită, baza de date este actualizată cu noua locație. La finalul procesului, după ce toate fișierele și dosarele au fost mutate, se actualizează dimensiunea dosarului țintă și a celui din care s-au mutat dosarele sau fișierele.



**J.** Adaugarea/stergerea unui fisier/dosar (de) la favorite

Procedura de adăugare sau ștergere din lista de favorite se realizează prin selectarea opțiunii corespunzătoare din meniul de opțiuni asociat unui fișier sau dosar. Atunci când un fișier sau dosar este deja inclus în lista de favorite, utilizatorul va observa doar opțiunea de ștergere, permițându-i să îl elimine din colecția sa de favorite. În situația inversă, dacă fișierul sau dosarul nu este încă marcat ca favorit, va fi disponibilă doar opțiunea de adăugare, oferind utilizatorului posibilitatea de a adăuga rapid și ușor elementul respectiv în lista sa de favorite.

**K.** Vizualizarea detaliilor al unui fisier/dosar

 În meniul de opțiuni, există o funcționalitate denumită "Detalii". Prin selectarea acestei opțiuni, se deschide un modal ce furnizează informații detaliate despre fișierul sau dosarul respectiv. Aceste date includ aspecte precum numele complet, numele original, data încărcării, ultima accesare, locația și altele. Utilizatorul poate accesa aceste informații pentru a obține o perspectivă detaliată asupra caracteristicilor și istoricului asociat fișierului sau dosarului selectat.

**L.** Descarcarea unui fisier/dosar

Descărcarea unui fișier este o acțiune directă în cadrul sistemului. Odată ce utilizatorul selectează această opțiune, se trimite o cerere către server pentru fișierul specific, acesta fiind validat și apoi transmis direct către utilizator.

În cazul descărcării unui dosar, procedura este similară celei pentru fișiere, cu excepția faptului că pe partea de server există un dosar in care se creaza dosarele care trbuie descarcate. Aceste dosare sunt apoi comprimate în fișiere zip, folosind biblioteca *archiver*, înainte de a fi transmise către utilizator. Această abordare permite eficientizarea transferului și gestionarea mai ușoară a mai multor elemente, precum dosare și subdosare, într-un format compact și ușor de descărcat.

**L.** Generarea unui link de descarcare

Această opțiune implică generarea unui link aleatoriu, realizată cu ajutorul bibliotecii *uuid*, care este valabil pentru o perioadă de 60 de minute. Orice persoană care dispune de acest link are posibilitatea de a descărca fișierul sau dosarul asociat. În cazul în care deja există un link generat pentru un anumit fișier sau dosar, utilizatorul poate șterge link-ul existent și genera unul nou în doar câteva secunde.

De asemenea, se oferă și posibilitatea de a efectua descărcarea prin scanarea unui cod QR asociat fișierului sau dosarului în cauză. Această funcționalitate asigură un proces eficient și flexibil pentru partajarea și accesarea conținutului în mod temporar și securizat.

**L.** Stergerea fisierelor/dosarelor

Ștergerea fișierelor sau dosarelor se efectuează din meniul de opțiuni disponibile pentru fiecare fișier sau dosar în parte. Atunci când utilizatorul optează pentru această acțiune, fișierul sau dosarul respectiv va fi transferat în coșul de gunoi. Dacă aceste elemente rămân în coșul de gunoi timp de 7 zile dupa care ele vor fi eliminate definitiv.

De asemenea, există posibilitatea de a recupera fișierele sau dosarele șterse înainte ca acestea să fie eliminate permanent. Utilizatorul are o fereastră de timp de 7 zile pentru a decide dacă dorește să recupereze sau să șteargă definitiv elementele aflate în coșul de gunoi.

**M.** Cautarea unui fisier/dosar

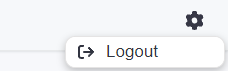
Utilizatorul are capacitatea de a efectua căutări specifice pe diverse pagini, cum ar fi "dashboard", "starred", "recents" sau "deleted". Pentru a localiza un fișier sau dosar specific, utilizatorul introduce prima dată pagina dorită (prin cuvinte-cheie de mai sus). Apoi, urmează să introducă numele exact al dosarului în care dorește să facă sortarea, cuvatul “all” pentru a cauta in toate dosarele din catgoria Dorita sau “/” pentru a cauta pe pagina, fara cauta in dosarele pe care le are. Dupa aceea urmeaza numele fisierului/dosarului dorit.

Pentru a face distincția între aceste opțiuni, ele sunt separate între ele prin separatorul "|" pentru a facilita identificarea și procesarea lor folosind metoda split.



**N.** Deconectarea

Procesul de deconectare de la cont se inițiază prin apăsarea butonului de setări și ulterior a butonului de "Log Out". La acest moment, se efectuează o cerere către server pentru a șterge token-ul de refresh asociat utilizatorului. După confirmarea ștergerii token-ului de refresh de către server, se procedează la ștergerea token-urilor din cookies de pe dispozitivul utilizatorului.

După ce token-urile au fost șterse cu succes, utilizatorul este redirecționat automat către pagina de autentificare. Această abordare asigură o deconectare completă și securizată, în care toate sesiunile active sunt încheiate și accesul la resursele protejate este imediat revocat.

Codul sursa: <https://github.com/Flavius2486/cloud_storage.git>

Aplicatia web poate fi vizualizata pe: <https://cloud.fl4v1u5.net>

Nume de utilizator: guestUPT

Parola: guestUPT1337