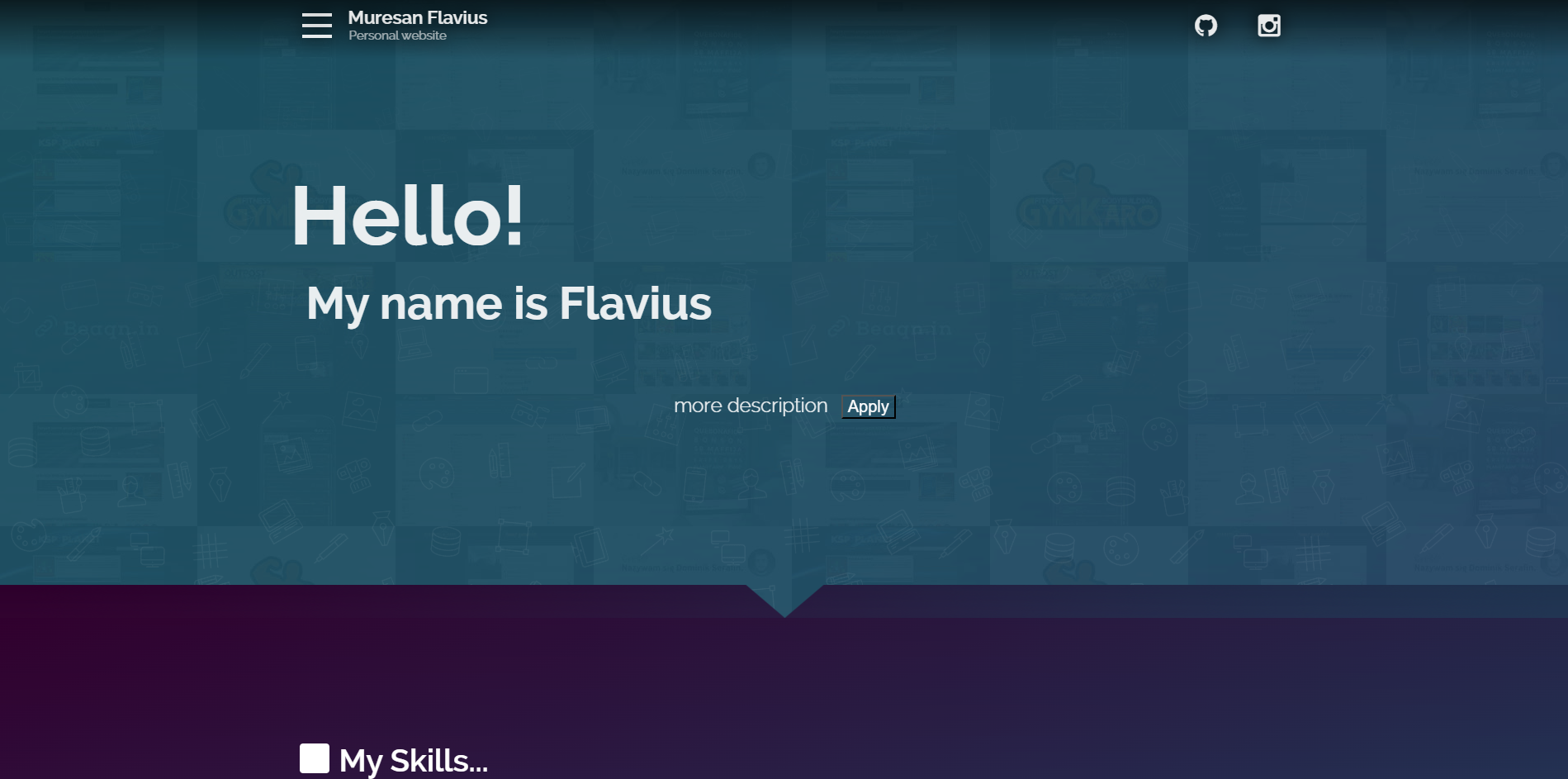
**Documantatie**

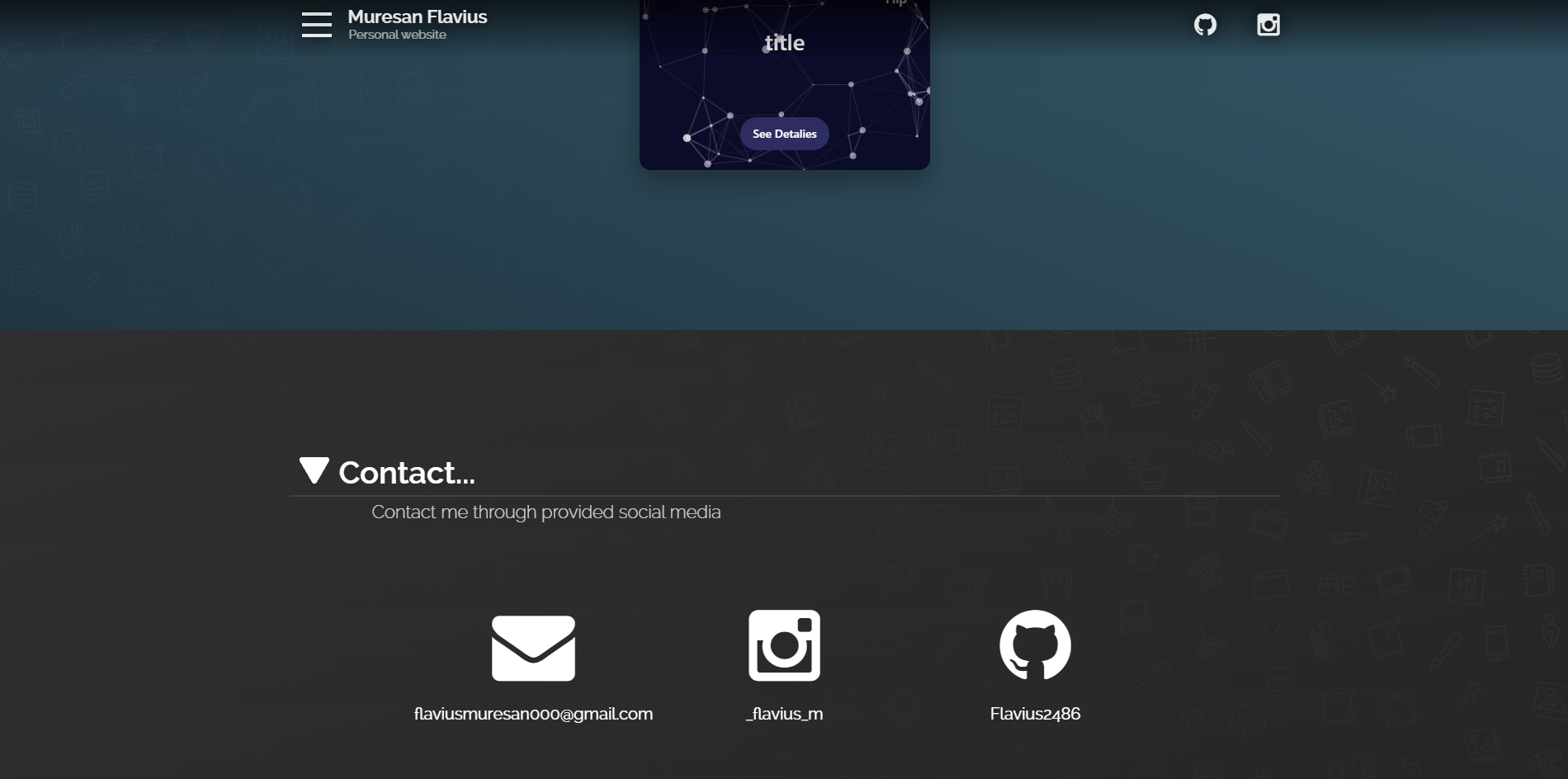
**1.SPECIFICAȚII GENERALE**

Acest proiect este un site personal, o aplicație web creată pentru a prezenta atât cunoștințele mele, cât și proiectele la care am lucrat în trecut.

Aplicația web începe cu o mică introducere despre mine, urmată de lista limbajelor de programare, sistemelor de operare, framework-urilor etc., pe care le cunosc. După aceasta, urmează o secțiune de proiecte prezentată sub formă de carduri. La click pe un card, utilizatorul este redirecționat către o altă pagină care conține detaliile proiectului (cum ar fi imagini, titlu, descriere etc.). După secțiunea de proiecte, urmează cea de contact, care conține adresa de email, linkurile către pagina mea de Instagram și GitHub.

De asemenea, există și o pagină de administrare, accesibilă cu un nume de utilizator/email si o parolă. Această pagină permite editarea numelui, introducerii și a altor informații cu un singur click pe paragraful corespunzător. Dacă textul este modificat, apare un buton "Aplicare". De asemenea, pe această pagină se poate adăuga/șterge limbaje de programare cunoscute, se pot edita linkurile (Instagram, GitHub, email) și se pot adăuga/șterge proiecte. Aceste modificări se fac printr-un popup care apare la clic pe butonul "Editare".





**2.LIMBAJE SI TEHNOLOGII FOLOSITE**

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.

MongoDB - bază de date open-source NoSQL scrisă în C++. Aceasta poate conține mai multe baze de date, colecții și indecși.

**3.ARHITECTURA APLICAȚIEI**

**A.** Arhitectura generală a aplicației

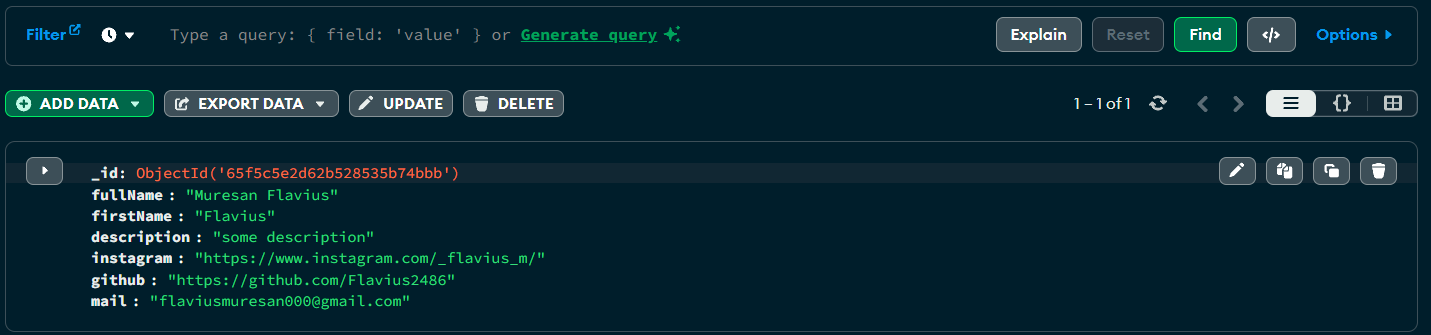
**Frontend**, sau partea client, într-o aplicație Nuxt este dezvoltată în handlebars și este responsabilă de interfața cu utilizatorul, de toate elementele pe care acesta le poate vedea și cu care poate interacționa. Această parte a aplicației se ocupă și de logica care rulează în browserul utilizatorului, inclusiv validarea datelor de intrare, manipularea evenimentelor și actualizarea interfeței grafice.

**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 MongoDB si poate sa modifice, sa stearga sau sa adauge date noi in baza de date. De asemenea, backend-ul se ocupa si de redirectionare, trimitand pagina Dorita can utilizatorul acceseaza o anumita ruta.

**B.** Structura bazei de date

Baza de date este formata 4 colectii: about\_me, admin, projects, skills.

Colectia about\_me este formata din 6 campuri: fullName(numele complet), firstName(prenumele), descrierea(introducerea de la inceputul aplicatiei), mail si link-urile de Instagram si de GitHub.



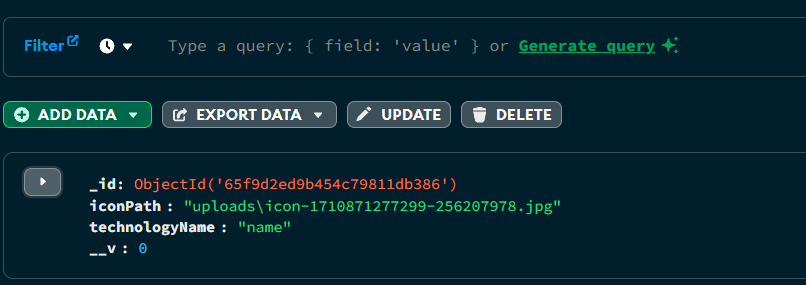
Colectia contine 2 campuri: numele de utilizator si parola.



Colectia projects este formata din 6 campuri: titlul proiectului, contributia, detalii, tehnologiile folosite, link-ul pentru codul sursa(daca este dat) si inca un camp de tip Array care contine locatia pozelor proiectului.



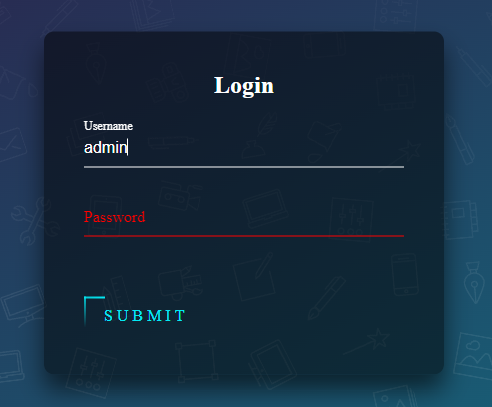
Colectia skills contine 2 campuri: numele liibajului de programare si sigla acestuia.



**4.DESCRIEREA FUNCTIONALITATIILOR APLICATIEI**

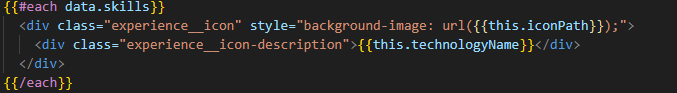
**A.** Autentificarea

Procesul de autentificare a administratorului se realizeaza cu express-session, un modul middleware în Express.js care facilitează crearea sesiunilor în aplicația web. Acesta stochează datele sesiunii pe server, oferind diverse opțiuni de stocare, și permite urmărirea activității unui utilizator între cereri. Aceasta sesioune expire 24 de ore de la autentificare.



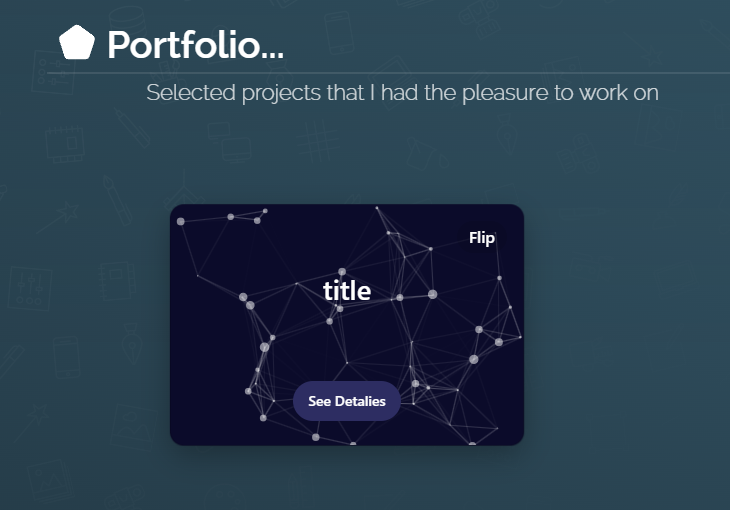
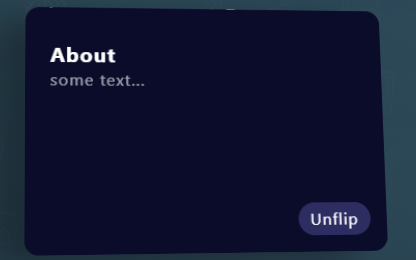
**B.** Afisarea datelor

La accesarea paginii, aplicația face o cerere către server, care extrage datele din baza de date MongoDB și le returnează sub formă de obiect JSON. După extragerea datelor, acestea sunt trimise către frontend împreună cu fișierul HBS corespunzător. După ce utilizatorul primește fișierul HBS și datele asociate, frontend-ul folosește metoda #each pentru a itera prin obiectul JSON și a crea elementele necesare. Odată ce obiectul JSON este primit de către frontend, acesta este utilizat pentru a actualiza interfața utilizatorului. De exemplu, în secțiunea skills se afișează atâtea sigle câte sunt în obiectul JSON.



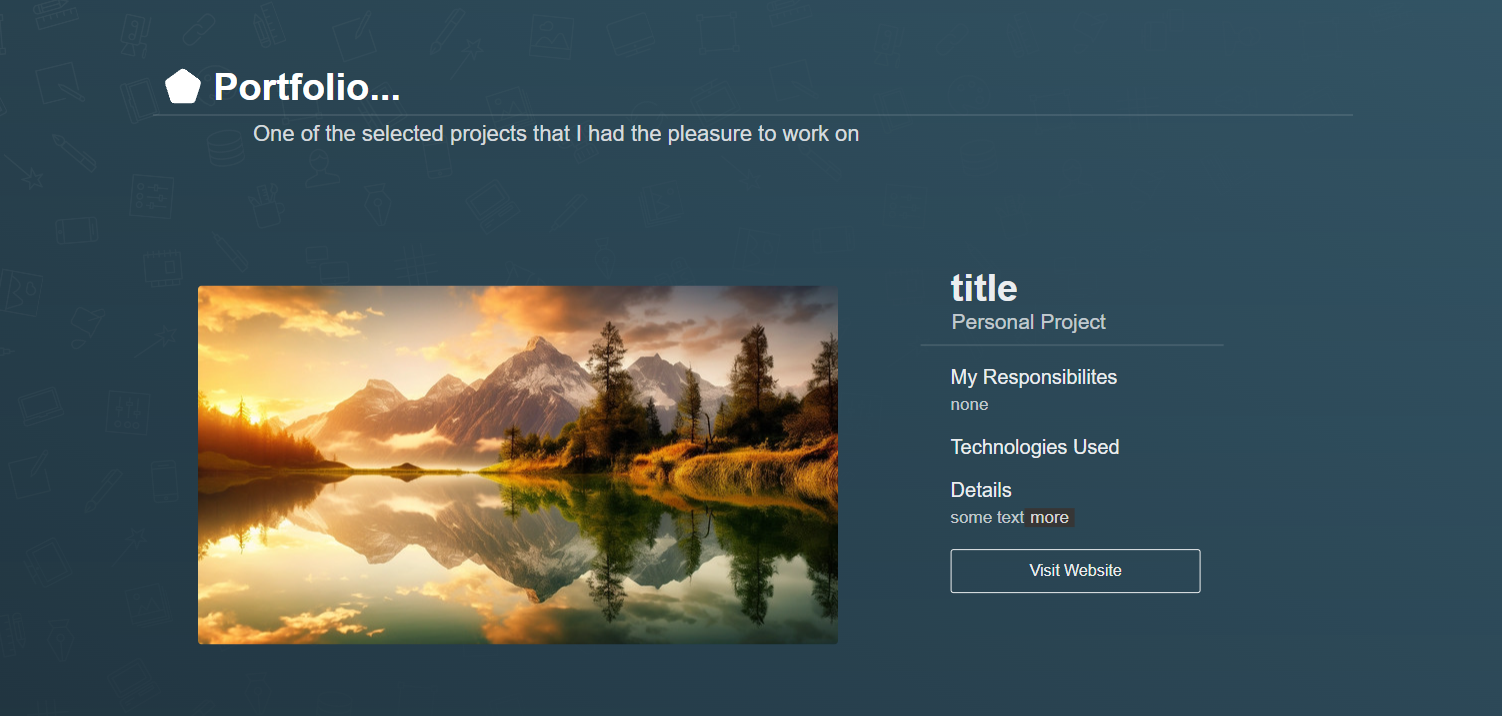
**C.** Vizualizarea unui proiec

În secțiunea de proiecte, fiecare proiect este reprezentat printr-un card 3D interactiv. Atunci când utilizatorul face clic pe un card, acesta este redirecționat către o pagină nouă, iar daca utilizatorul doreste sa vada o mica descriere poate sa faca click pe butonul “Flip” care intoarce cardul 180 de grade. În URL-ul acestei noi pagini, este inclus ID-ul unic al proiectului, generat automat de MongoDB la momentul inserării proiectului în baza de date.

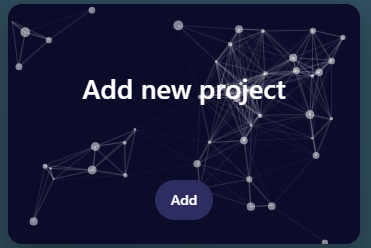
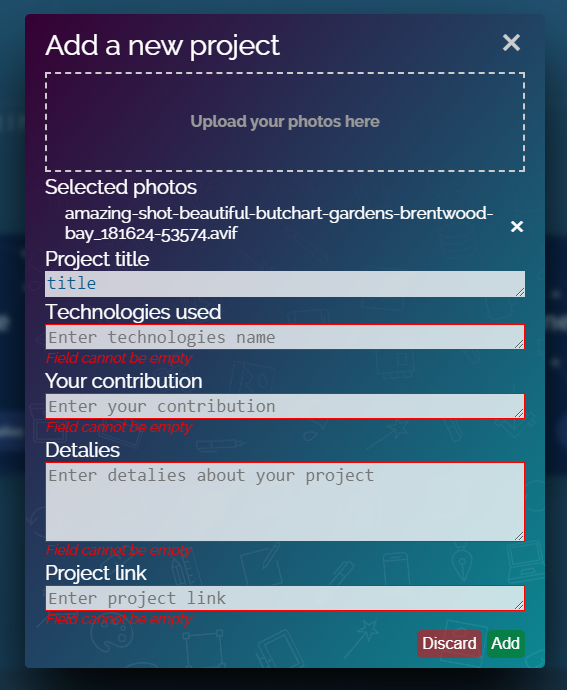
Pe această pagină nouă, serverul primește acest ID și efectuează o interogare către baza de date MongoDB pentru a obține toate detaliile asociate proiectului cu ID-ul respectiv. Aceste detalii includ titlul proiectului, descrierea, limbajels si tehnologiils folosite, link-ul catre codul sursa etc.

După ce datele sunt extrase din baza de date, acestea sunt procesate și prezentate într-un format corespunzător pe pagina HBS (Handlebars). Variabilele asociate cu detaliile proiectului sunt atașate șablonului HBS pentru a fi afișate corect în HTML-ul generat. Astfel, utilizatorul poate vedea informații detaliate despre proiectul selectat.



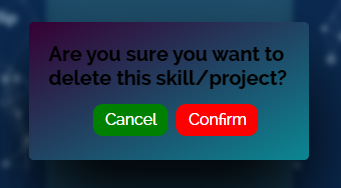
**D.** Adaugarea unui proiect

Adăugarea unui proiect se face de pe pagina de administrator făcând clic pe cardul cu titlul "Adaugă un proiect nou", care va deschide un popup în care se încarcă imaginile proiectului. Administratorul introduce titlul, contribuția, tehnologiile și limbajele folosite, descrierea și link-ul către codul sursă în acest popup. La apăsarea butonului "Adaugă", se verifică dacă toate câmpurile au fost completate; în caz contrar, câmpurile necompletate sunt marcate cu roșu. După validarea datelor, acestea sunt trimise către backend unde sunt introduse în baza de date, iar imaginile proiectului sunt stocate pe disc cu ajutorul bibliotecii Multer. După stocarea datelor, pagina este reîmprospătată.

****

**E.** Stergerea unui proiect

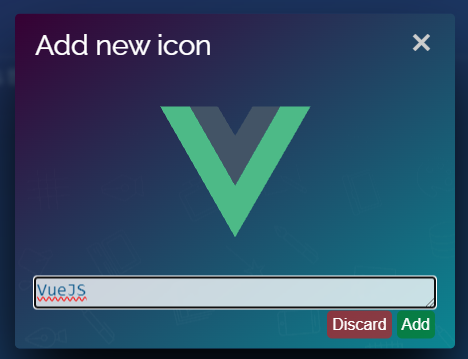
Ștergerea unui proiect se face de pe pagina de administrator făcând dublu clic pe cardul care reprezintă proiectul ce trebuie șters. Pentru a confirma sau anula ștergerea, se apasă pe butoanele corespunzătoare din popup-ul care va apărea. În caz afirmativ, se face o cerere către backend trimitând id-ul acestuia. Datele proiectului sunt șterse din baza de date, după care imaginile acestuia sunt șterse de pe disc folosind biblioteca fs. Apoi se reîmprospătează pagina.

****

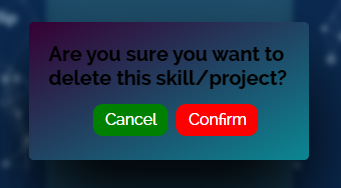
**E.** Adaugarea/stergerea unui skill

Aceste acțiuni se pot realiza numai de pe pagina de administrator.

Adăugarea unui skill se face făcând clic pe ultimul afișat, similar cu un plus. La apăsarea acestuia, se va deschide un popup în care se introduce numele și sigla acestuia. La apăsarea butonului "Adaugă", imaginea și numele sunt trimise către backend, unde sunt memorate în baza de date și pe disc (imaginea).

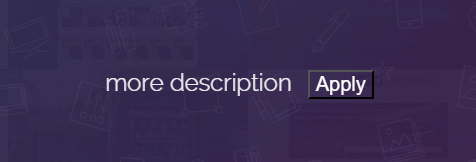
 

Ștergerea unui skill se realizează la apăsarea acestuia și confirmarea ștergerii (la fel ca la ștergerea unui proiect).

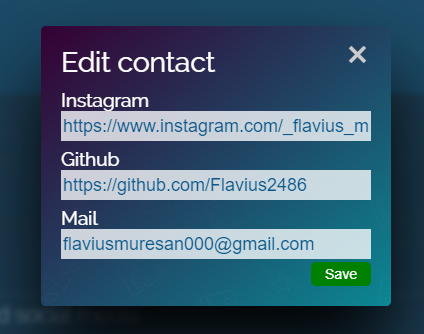
****

**F.** Editarea informatiilor personale

La accesarea paginii de administrator, informațiile personale pot fi modificate cu ușurință făcând clic pe textul care trebuie modificat. La orice modificare efectuată, va apărea un buton pentru a salva modificările făcute. La apasarea butonului “Apply” se trimt datele modificate la server si sunt salvate in baza de date.



Pentru a modifica link-urile de contact, se apasă pe butonul "Editare" din secțiunea respectivă. Va apărea o fereastră pop-up care permite administratorului să modifice aceste link-uri. La apasarea butonului “Save” se trimit link-ruile la server dupa care se actualizeaza baza de date.



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

Aplicatia web poate fi vizualizata pe:

<https://fl4v1u5.net>