

Sprint Restanta

Perioada: 03.02.2024-18.02.2024

Disponibilitate:

- ★ 05.02.2024: Intreaga echipa a hotarat noul feature si functionalitatile necesare pentru acest sprint si am discutat despre viitoarele teste care vor fi implementate. Am adaugat noi user stories si am actualizat ReadMe-ul. De asemenea am hotarat taskurile si disponibilitatea fiecarui membru spre realizarea acestora;
- ★ 11.02.2024 - 12.02.2024: Bojneagu David a lucrat la realizarea testelor end-to-end si integration pe partea de backend;
- ★ 12.02.2024: Gheorghe Cosmina a realizat Backendul pentru salvari(model, rute, controller) pentru ca mai apoi Neagu Mara sa poata face Frontendul, avand disponibilitate in aceeasi zi;
- ★ 16.02.2024 - 17.02.2024: Georgescu Elena a adaugat functionalitatea ca mesterul sa poata vedea cati oameni i-au salvat postarea atat pe backend cat si frontend.
- ★ 18.02.2024: Toata echipa a lucrat la rapoarte: Bojneagu David si Talipga Vlad au realizat Software Architecture, iar Georgescu Elena, Gheorghe Cosmina si Neagu Mara au facut Sprint Reportul.

User Stories:

1. "In calitate de client, vreau sa pot salva o postare care ma intereseaza pentru o viitoare programare."
2. "In calitate de client, vreau sa pot accesa o pagina de unde sa vad postarile pe care le-am salvat."
3. "In calitate de mester, vreau sa am o evidenta a numarului de persoane care sunt interesate de o postare facuta de mine."

Acceptance Criteria:

1. Clientul salveaza o postare FIINDCA aceasta reprezinta un punct de interes, iar CAND va dori sa faca o viitoare programare ATUNCI acesta o poate gasi mai usor.
2. Clientul acceseaza pagina de salvari, FIINDCA postarile salvate se afla pe o pagina dedicata, iar CAND vrea sa faca o programare ATUNCI va gasi anuntul dorit.
3. Mesterul vede numarul de salvari ale unei postari FIINDCA aplicatia tine o evidenta a acestora, astfel in momentul CAND vrea sa vada serviciile de care sunt atrasi clientii ATUNCI acest lucru poate fi observat in respectiva postare.

Planned items and their status:

- **Proposed:**
 1. Integration Testing pentru frontend si backend;
 2. End-to-end Testing pentru frontend si backend;
 3. Realizare modelului pentru salvari;
 4. Realizarea functiilor tip CRUD si rutelor corespunzatoare salvarilor;
 5. Afisarea butonului de salvare;
 6. Realizarea frontendului pentru pagina de salvari;
 7. Afisarea numarului de salvari ale unei postari;
 8. Eliminarea butonului de like, inlocuirea sa cu butonul de salvare.
- **Completed:**
 1. Integration Testing pentru backend;
 2. End-to-end Testing pentru backend;
 3. Realizare modelului pentru salvari;
 4. Realizarea functiilor tip CRUD si rutelor corespunzatoare salvarilor;
 5. Afisarea butonului de salvare;
 6. Realizarea frontendului pentru pagina de salvari;
 7. Afisarea numarului de salvari ale unei postari doar daca postarea a fost realizata de userul logat;
 8. Eliminarea butonului de like, inlocuirea sa cu butonul de salvare.
- **Removed before the sprint end:**
 1. Integration Testing pentru frontend ;
 2. End-to-end Testing pentru frontend.

Done Items:

- Realizarea salvarilor postarii indeplineste acest **Acceptance Criteria** "Clientul salveaza o postare FIINDCA aceasta reprezinta un punct de interes, iar CAND va dori sa faca o viitoare programare ATUNCI acesta o poate gasi mai usor."
- Realizarea paginii de afisare a postarilor salvate indeplineste acest **Acceptance Criteria** : "Clientul acceseaza pagina de salvari, FIINDCA postarile salvate se afla pe o pagina dedicata, iar CAND vrea sa faca o programare ATUNCI va gasi anuntul dorit."
- Aduagarea numarului de persoane care au salvat o postare indeplineste acest **Acceptance Criteria** : "Mesterul vede numarul de salvari ale unei postari FIINDCA aplicatia tine o evidenta a acestora, astfel in momentul CAND vrea sa vada serviciile de care sunt atrasi clientii ATUNCI acest lucru poate fi observat in respectiva postare."

Sprint Review

Difficulties Encountered:

- Pe partea de frontend, m-am concentrat pe realizarea testelor end-to-end și de integrare. În cadrul testelor de integrare, mi-am propus să verific dacă pagina principală este randată corect și dacă butonul de pornire a tutorialului este prezent. Pentru aceasta, era necesar să randez întreaga pagină de start, inclusiv toate elementele sale. Inițial, am încercat să folosesc un store Redux, dar deoarece acesta nu a fost configurat inițial, am întâmpinat dificultăți în setarea sa acum. Astfel, am decis să renunț la această abordare. Am încercat inițial să utilizez funcția `shallow`, dar am constatat că aceasta nu randează profund pagina de start, ceea ce explică de ce butonul de pornire nu era afișat. Ulterior, am descoperit că era necesar să utilizez funcția `mount` împreună cu un store Redux, despre care am menționat anterior că nu a fost configurat cu succes.
- Pe partea de backend, am adoptat o nouă abordare. Prima problemă întâmpinată a fost setarea variabilelor de mediu în fluxul de lucru al acțiunilor GitHub. După o cercetare, am descoperit că pot utiliza GitHub Secrets pentru a seta aceste variabile de mediu și apoi să le accesez în fluxul meu de integrare continuă. Inițial, am încercat să simulez un proces de înregistrare, dar am realizat că trebuie să șterg înregistrarea imediat după aceasta pentru a evita problemele la fiecare rulare a fluxului de lucru. Testele se executau corect, dar am observat că aveam nevoie de un indicator pentru a specifica în mod explicit ieșirea din comanda de testare. De asemenea, a trebuit să specific un timeout maxim pentru fluxul de lucru al testelor pentru a evita rularea lor la nesfârșit.
- În ceea ce privește instalarea pachetelor, am întâmpinat o problemă în ceea ce privește instalarea modulului `bcrypt`, care, din motive neclare, nu a putut fi instalat decât separat.
- Pentru pagina separată de salvări, am încercat să implementez un stil de randare a postărilor folosind un aspect de tip Masonry. După o cercetare mai amplă, am descoperit două modalități de a importa acest aspect de layout: `"react-masonry-css"` și `"SavedPage.css"`. Cu toate acestea, am constatat că niciuna dintre aceste metode nu randează corect layout-ul respectiv. Astfel, pentru a evita complicațiile, am optat să afișez postările una sub alta, adoptând o opțiune mai sigură și mai convențională.

Sprint Retrospective

Collaboration and Team Performance:

- Echipa noastră a realizat un efort colectiv, ce s-a manifestat printr-o colaborare armonioasă și o comunicare eficientă, care au condus la finalizarea cu succes a task-urilor stabilite la începutul sprint-ului. În cadrul unei întâlniri desfășurate pe 5 februarie, am stabilit în mod detaliat sarcinile individuale ale fiecărui membru și estimările de timp pentru rezolvarea acestora. Având în vedere că task-ul legat de salvări a fost împărțit în trei etape distincte, fiecare membru a depins de contribuția și progresul celorlalți. Comunicarea fluentă a fost importanta pentru acest proces, permițând soluționarea rapidă a oricăror neclarități referitoare la funcționalitățile specifice ale controller-ului de salvări. Ca rezultat, progresul a fost liniar și eficient, iar fiecare membru și-a îndeplinit responsabilitățile cu seriozitate și determinare. În concluzie, succesul echipei noastre a fost rezultatul unui efort colectiv susținut și al unei abordări profesioniste în gestionarea task-urilor.

What was easier than expected:

- Implementarea a fost mult mai rapida deoarece ne-am obisnuit cu workflowul tehnologiilor utilizate. Din sprinturile trecute am observat ce tipuri de taskuri se potrivesc fiecarui membru astfel incat impartirea acestora a fost mult mai eficienta in cadrul noului sprint.

Reflection on Sprint Achievements:

- Prin realizarea featureului nou de salvari din acest sprint, aplicatia noastra a ajuns intr-o forma finala de dezvoltare, marcand toate dorintele pe care le aveam la inceput de drum.