We are having a problem billing the locqum organization. Please update your payment method or call your payment provider for details on why the transaction failed.
You can always contact support with any questions.

Commit fe7a7cc

claude committed last month Verified

```
fix: persist auth tokens in localStorage to survive payment redirects

Tokens were stored only in memory (signals), so navigating to the
PlaceToPay payment gateway and returning caused a full page reload
that wiped the session. Now tokens are saved to localStorage on login/
register and restored on app initialization, with automatic profile
loading and token refresh fallback.

https://claude.ai/code/session_01WTbEJHa9nd8kVVoKCirXH8
```

master

1 parent 3759a9c commit fe7a7cc

1 file changed +28 -1 lines changed

Top

Filter files...

frontend/src/app/core/auth

auth.service.ts

1 file changed +28 -1 lines changed

Search within code

frontend/src/app/core/auth/auth.service.ts

```
@@ -7,6 +7,9 @@ import { AuthResponse, UserProfile } from './auth.models';
7  7
8  8   @Injectable({ providedIn: 'root' })
9  9   export class AuthService {
   10  +   private static readonly ACCESS_TOKEN_KEY = 'auth_access_token';
```

```
                +   private static readonly REFRESH_TOKEN_KEY = 'auth_refresh_token';
                +
10  13             private readonly accessToken = signal<string | null>(null);
11  14             private readonly refreshTokenValue = signal<string | null>(null);
12  15             private readonly user = signal<UserProfile | null>(null);
                @@ -18,7 +21,27 @@ export class AuthService {
18  21             constructor(
19  22               private readonly http: HttpClient,
20  23               private readonly router: Router,
21      -          ) {}
                +  ) {
                +    this.restoreSession();
                +  }
                +
                +  private restoreSession(): void {
                +    const accessToken = localStorage.getItem(AuthService.ACCESS_TOKEN_KEY);
                +    const refreshToken = localStorage.getItem(AuthService.REFRESH_TOKEN_KEY);
                +    if (accessToken && refreshToken) {
                +      this.accessToken.set(accessToken);
                +      this.refreshTokenValue.set(refreshToken);
                +      this.loadProfile().subscribe({
                +        error: () => {
                +          // Token expired or invalid — try refreshing
                +          this.refreshToken().subscribe({
                +            next: () => this.loadProfile().subscribe(),
                +            error: () => this.logout(),
                +          });
                +        },
                +      });
                +    }
                +  }
22  45
23  46             getAccessToken(): string | null {
24  47               return this.accessToken();
                @@ -63,11 +86,15 @@ export class AuthService {
63  86               this.accessToken.set(null);
64  87               this.refreshTokenValue.set(null);
65  88               this.user.set(null);
                +    localStorage.removeItem(AuthService.ACCESS_TOKEN_KEY);
                +    localStorage.removeItem(AuthService.REFRESH_TOKEN_KEY);
66  91               this.router.navigate(['/']);
```

```
67   92       }
68   93
69   94       private setTokens(res: AuthResponse): void {
70   95         this.accessToken.set(res.accessToken);
71   96         this.refreshTokenValue.set(res.refreshToken);
     97   +     localStorage.setItem(AuthService.ACCESS_TOKEN_KEY, res.accessToken);
     98   +     localStorage.setItem(AuthService.REFRESH_TOKEN_KEY, res.refreshToken);
72   99       }
73   100    }
```

Comments 0

Lock conversation

Comment

Subscribe You're not receiving notifications from this thread.