**Detailná analýza sekcie registrácie - Swaply projekt**

**1. Funkcie, triedy a endpointy súvisiace s registráciou**

**Hlavné endpointy:**

* **POST /api/accounts/auth/register/ - Registrácia nového používateľa**
* **POST /api/accounts/auth/verify-email/ - Overenie email adresy**
* **POST /api/accounts/auth/resend-verification/ - Znovu odoslanie verifikačného emailu**
* **GET /api/accounts/auth/check-email/<email>/ - Kontrola dostupnosti emailu**

**Kľúčové triedy a funkcie:**

**Models (backend/accounts/models.py):**

* **User(AbstractUser) - Rozšírený User model s dodatočnými poľami**
* **UserProfile - Rozšírené informácie o používateľovi**
* **EmailVerification - Model pre email verifikáciu s UUID tokenmi**

**Views (backend/accounts/views/auth.py):**

* **register\_view() - Hlavná registračná funkcia s rate limiting**
* **verify\_email\_view() - Overenie email tokenu**
* **resend\_verification\_view() - Znovu odoslanie verifikačného emailu**
* **check\_email\_availability\_view() - Kontrola dostupnosti emailu**

**Serializers (backend/accounts/serializers.py):**

* **UserRegistrationSerializer - Validácia a spracovanie registračných údajov**
* **EmailVerificationSerializer - Validácia verifikačných tokenov**
* **ResendVerificationSerializer - Validácia pre znovu odoslanie**

**2. Účel a flow jednotlivých funkcií**

**register\_view():**

* **Účel: Vytvorenie nového používateľa s email verifikáciou**
* **Flow:**

1. **Validácia vstupných údajov cez serializer**
2. **Vytvorenie používateľa s is\_verified=False**
3. **Vytvorenie UserProfile a EmailVerification záznamu**
4. **Odoslanie verifikačného emailu**
5. **Audit logovanie úspešnej registrácie**

**verify\_email\_view():**

* **Účel: Overenie email adresy pomocou UUID tokenu**
* **Flow:**

1. **Validácia tokenu (expirovanie, použitie)**
2. **Označenie používateľa ako overený (is\_verified=True)**
3. **Generovanie JWT tokenov pre automatické prihlásenie**
4. **Audit logovanie verifikácie**

**UserRegistrationSerializer.create():**

* **Účel: Vytvorenie používateľa s bezpečnostnými opatreniami**
* **Flow:**

1. **Validácia hesiel a business logiky**
2. **Vytvorenie používateľa s is\_verified=False**
3. **Vytvorenie profilu a verifikačného tokenu**
4. **Odoslanie emailu**

**3. Bezpečnostné riziká a analýza**

**✅ Silné stránky:**

**Validácia vstupov:**

* **Komplexné custom validátory v validators.py**
* **SQL injection ochrana v SecurityValidator.validate\_input\_safety()**
* **XSS ochrana cez regex patterns**
* **Email validácia s regex a dĺžkovými limitmi**
* **Heslo validácia s komplexitou (veľké/malé písmená, čísla)**
* **Veková validácia (min. 13 rokov)**

**Rate limiting:**

* **Implementovaný v rate\_limiting.py**
* **Rôzne limity pre rôzne akcie:**
* **Registrácia: 3 pokusy/15 min, blokovanie 30 min**
* **Prihlásenie: 5 pokusov/15 min, blokovanie 60 min**
* **Email verifikácia: 5 pokusov/15 min, blokovanie 60 min**

**Audit logovanie:**

* **Komplexný audit systém v audit\_logger.py**
* **Logovanie všetkých bezpečnostných udalostí**
* **IP adresa a user agent tracking**

**JWT bezpečnosť:**

* **Token rotation (ROTATE\_REFRESH\_TOKENS: True)**
* **Blacklist po rotácii (BLACKLIST\_AFTER\_ROTATION: True)**
* **60 min access token, 7 dní refresh token**

**⚠️ Stredné riziká:**

**Email verifikácia:**

* **Token expiruje po 24 hodinách (možno príliš krátko)**
* **Žiadna ochrana proti enumeration attack na email check endpoint**
* **Chýba rate limiting na email check endpoint**

**Heslo:**

* **Používa Django štandardné validátory (dobré, ale základné)**
* **Chýba kontrola proti dictionary attack**
* **Žiadne zložité požiadavky na heslo (napr. špeciálne znaky)**

**Session management:**

* **Používa JWT namiesto session cookies**
* **Refresh token má 7 dní platnosť (možno príliš dlho)**

**🔴 Vysoké riziká:**

**CSRF ochrana:**

* **CSRF middleware je zapnutý, ale pre API endpointy sa nepoužíva**
* **Chýba CSRF token validácia v API calls**

**CORS konfigurácia:**

* **CORS\_ALLOW\_CREDENTIALS = True môže byť rizikové**
* **Špecifické origins sú nastavené, ale môžu byť zraniteľné**

**Error handling:**

* **V DEBUG=True sa zobrazujú detailné chybové správy**
* **Môže viesť k informačnému úniku**

**4. Hodnotenie best practices**

**✅ Dodržané best practices:**

* **Použitie Django ORM (žiadne raw SQL)**
* **Centralizované validátory**
* **Proper error handling s custom exception handler**
* **Audit logovanie**
* **Rate limiting**
* **Environment variables pre konfiguráciu**
* **Proper password hashing (Django default)**

**❌ Chýbajúce best practices:**

* **Chýba input sanitization pre HTML content**
* **Žiadna ochrana proti timing attacks**
* **Chýba rate limiting na email check endpoint**
* **Žiadne CAPTCHA pre registráciu**
* **Chýba monitoring a alerting pre bezpečnostné udalosti**

**5. Konkrétne vylepšenia pre bezpečnosť**

**Vysoká priorita:**

1. **Pridať CAPTCHA pre registráciu a email verifikáciu**
2. **Implementovať rate limiting na email check endpoint**
3. **Rozšíriť heslo validáciu o špeciálne znaky**
4. **Pridať CSRF token pre API calls**
5. **Implementovať account lockout po viacerých neúspešných pokusoch**

**Stredná priorita:**

1. **Rozšíriť token expiraciu na 48 hodín pre email verifikáciu**
2. **Pridať monitoring pre bezpečnostné udalosti**
3. **Implementovať 2FA pre kritické operácie**
4. **Pridať input sanitization pre HTML content**
5. **Implementovať session management s proper logout**

**Nízka priorita:**

1. **Pridať password strength meter v UI**
2. **Implementovať social login (Google OAuth je vypnutý)**
3. **Pridať email templates pre lepšiu UX**
4. **Implementovať progressive registration (krok za krokom)**

**6. Zhodnotenie stavu bezpečnosti**

**Celkové hodnotenie: STREDNÉ RIZIKO ⚠️**

**Dôvody:**

* **Pozitíva: Dobrá základná bezpečnosť, rate limiting, audit logovanie, proper validácia**
* **Negatíva: Chýba CAPTCHA, CSRF ochrana pre API, príliš dlhé refresh tokeny, chýba monitoring**

**Odporúčania:**

1. **Okamžite implementovať CAPTCHA pre registráciu**
2. **Pridať rate limiting na email check endpoint**
3. **Skrátiť refresh token lifetime na 24 hodín**
4. **Implementovať proper CSRF ochranu pre API**
5. **Pridať monitoring a alerting pre bezpečnostné udalosti**

**Kód je napísaný podľa Django best practices s dobrými bezpečnostnými základmi, ale potrebuje dodatočné ochranné mechanizmy pre produkčné nasadenie.**

1. **Pridať CAPTCHA pre registráciu a email verifikáciu → ➡️ treba test (registrácia s/bez captcha).**
2. **Rate limiting na email check endpoint → ➡️ treba test (veľa requestov, posledný sa zamietne).**
3. **Rozšíriť heslo validáciu o špeciálne znaky → ➡️ treba test (heslo bez znaku = chyba, heslo so znakom = úspech).**
4. **Pridať CSRF token pre API calls → ➡️ treba test (request bez CSRF tokenu = zamietnutý).**
5. **Account lockout po viacerých neúspešných pokusoch → ➡️ treba test (opakované zlé heslo = lockout).**

**Stredná priorita**

1. **Predĺžiť expiraciu email tokenu na 48 hodín → ❌ netreba test (stačí nastavenie).**
2. **Monitoring bezpečnostných udalostí → ❌ netreba test (konfiguračná vec).**
3. **Implementovať 2FA pre kritické operácie → ➡️ treba test (login bez 2FA = zamietnuté, login s 2FA = prejde).**
4. **Input sanitization pre HTML content → ➡️ treba test (vložený <script> sa odfiltruje).**
5. **Proper session management s logoutom → ➡️ treba test (po logout request už neprejde).**

**Nízka priorita**

1. **Password strength meter v UI → ❌ netreba test (iba frontend).**
2. **Social login (Google OAuth) → ➡️ treba test (login prebehne správne, zlý token = chyba).**
3. **Email templates pre lepšiu UX → ❌ netreba test (vizuálna vec).**
4. **Progressive registration (krok za krokom) → ➡️ treba test (kroky musia ísť v poradí, preskočenie = zamietnuté).**

**3. �� Autocomplete: -** toto treba dať do sekcie vyhladavanie

*// Pre mestá a firmy*