

MediaMarket - zadání pro vytvoření backendu

1) Cíl a rozsah

Postavit backend tak, aby fungovaly všechny klíčové flow:

1. Registrace (Médium / Agentura) + ověření e-mailem
2. Přihlášení/odhlášení + obnova hesla (nutné pro reálný provoz)
3. Médium: správa nabídek (create/list/detail/edit/archivace)
4. Agentura: objednávka/poptávka na nabídku (včetně výpočtů a povinného "finálního klienta" dle nastavení nabídky)
5. Dashboardy: médium/agentura vidí své objednávky + stav; médium stav mění
6. E-mailové notifikace pro obě strany
7. Admin: přehled všech objednávek, filtry, výpočet provizí 2,5 % / 5 %, export do fakturační šablony

2) Role a oprávnění (RBAC)

Role:

- **VISITOR** (nepřihlášený) – typicky může prohlížet jen homepage, ale nemůže vidět výpis nabídek a detail nabídek a nemůže objednávat.
- **AGENCY** – může objednávat nabídky, vidí svoje poptávky/objednávky.
- **MEDIA** – může vytvářet nabídky, vidí objednávky na svoje nabídky, mění jejich stav.
- **ADMIN** – vše, navíc provize/exporty, správa uživatelů.

Zásadní pravidla:

- Agentura **nesmí** vytvářet nabídky.

- Médium **nesmí** objednávat.
- Stav objednávky může měnit **jen médium**, kterému patří nabídka nebo **admin**.

3) Datový model (doporučené tabulky)

Níže je návrh minimální DB. Developer musí porovnat s existujícími poli ve frontendu a doplnit vše, co ve formulářích reálně je.

3.1 Users (auth)

- `id` (UUID)
- `email` (unique)
- `password_hash` / řešit přes provider (Supabase/Auth0/NextAuth)
- `email_verified_at`
- `role` ENUM(AGENCY,MEDIA,ADMIN)
- `created_at, last_login_at`
- volitelně: `status` ENUM(ACTIVE,SUSPENDED,PENDING) – hodí se pro “ověřená média”

3.2 CompanyProfile (společnost)

- `id` (UUID)
- `user_id` (FK Users)
- `company_name` (Firma)
- `contact_name` (Kontaktní osoba)
- `phone`
- `type` ENUM(AGENCY,MEDIA) (redundantně k roli – ale praktické)
- `is_verified` (bool) – pokud chcete “ověřená média/partneři”

- `gdpr_consent_at` (timestamp) – souhlas se zpracováním osobních údajů při registraci
- `created_at, updated_at`

3.3 Offers (nabídky)

Pole vycházejí z popisu + prostor pro existující UI:

- `id` (UUID)
- `media_company_id` (FK CompanyProfile)
- `title / format_name` (podle UI)
- `description`
- **Cena model**
 - `pricing_model` ENUM(`UNIT_PRICE,CPT`)
 - `unit_price` (decimal, nullable)
 - `cpt` (decimal, nullable)
 - `min_order_value` (decimal, nullable) – “Minimální hodnota objednávky”
- **Tagy**
 - `is_last_minute` bool
 - `is_special` bool
 - `is_promo` bool
- **Termíny**
 - `deadline_assets_at` (date) – “Deadline na dodání podkladů”
 - `last_order_day` (date) – “Poslední možný den objednání”
- **Technické podmínky**

- `technical_conditions_text` (text, nullable)
 - `technical_conditions_url` (text/url, nullable)
 - `technical_conditions_file_id` (FK OfferFiles, nullable) – PDF upload
- **Finální klient**
 - `requires_final_client` bool
 - `status` ENUM(DRAFT,PUBLISHED,ARCHIVED) – doporučeno kvůli mazání
 - `created_at`, `updated_at`

3.4 OfferFiles (soubory k nabídce)

- `id`
- `offer_id`
- `file_type` ENUM(Technical CONDITIONS)
- `storage_path / url`
- `mime_type`, `size`
- `created_at`

3.5 Orders (objednávky/poptávky)

- `id` (UUID)
- `order_number` (unikátní, human-friendly; např. MMH-2026-000123)
- `offer_id` (FK Offers)
- `agency_company_id` (FK CompanyProfile)
- `media_company_id` (FK CompanyProfile) – denormalizace kvůli rychlým filtrům
- `preferred_from` (date)

- `preferred_to` (date)
- **Cenové vstupy**
 - `pricing_model_snapshot` ENUM(UNIT_PRICE,CPT) – snapshot z nabídky
 - `unit_price_snapshot` (decimal, nullable)
 - `cpt_snapshot` (decimal, nullable)
 - `quantity_units` (int, nullable) – Počet ks (1–100)
 - `impressions` (int, nullable) – Počet zobrazení
 - `total_price` (decimal) – **počítat na backendu**, ne věřit frontendu
- `final_client` (text, nullable; povinné pokud `requires_final_client`)
- `note` (text)
- `status` ENUM(NEW,IN_PROGRESS,CLOSED) – default NEW
- `created_at, updated_at`
- `status_changed_at, status_changed_by` (user_id) – audit

3.6 OrderStatusHistory (audit – doporučeno)

- `id`
- `order_id`
- `old_status, new_status`
- `changed_by` (user_id)
- `changed_at`

3.7 CommissionBatches (admin – provize)

- `id`
- `created_by` (admin user_id)
- `commission_rate` ENUM(`0.025,0.05`)
- `period_from, period_to` (nebo jen seznam objednávek)
- `created_at`

3.8 CommissionBatchItems

- `id`
- `batch_id`
- `order_id`
- `order_total_price`
- `commission_amount`

4) API / Backend funkce (kontrakty)

Může být REST nebo GraphQL. Níže REST návrh:

4.1 Auth

- `POST /auth/register`
Vstup: firma, kontaktní osoba, email, telefon, role (MEDIA/AGENCY),
`gdpr_consent=true`
Akce: vytvořit user + company profile, poslat verifikační email.
- `POST /auth/verify-email` (token)
- `POST /auth/login`
- `POST /auth/logout`
- `POST /auth/forgot-password`

- POST /auth/reset-password

4.2 Offers (MEDIA)

- POST /offers (MEDIA)
- GET /offers (listing; veřejný nebo dle UI)
- GET /offers/:id
- PATCH /offers/:id (jen vlastník média)
- POST /offers/:id/publish / POST /offers/:id/archive
- POST /offers/:id/files (upload PDF pro technické podmínky)

Validace nabídky:

- `pricing_model` musí být buď UNIT_PRICE nebo CPT.
- Pokud UNIT_PRICE → `unit_price` required; `quantity_units` v objednávce required.
- Pokud CPT → `cpt` required; `impressions` v objednávce required.
- Tagy jsou nezávislé checkboxy.
- `last_order_day` nesmí být po `deadline_assets_at` (pokud obě vyplněné).

4.3 Orders (AGENCY + MEDIA)

- POST /offers/:id/orders (AGENCY)
Backend:

- z nabídky si udělá snapshot ceny/modelu,
- zkонтroluje povinné `final_client`,
- spočítá `total_price`:
 - UNIT_PRICE: `unit_price * quantity_units`

- CPT: (`impressions/1000`) * cpt
 - vygeneruje `order_number`,
 - uloží, pošle e-maily oběma stranám.
- `GET /agency/orders` (AGENCY) – moje objednávky
- `GET /media/orders` (MEDIA) – objednávky na moje nabídky
- `GET /orders/:order_number` nebo `/orders/:id` (role-based)
- `PATCH /media/orders/:id/status` (MEDIA)
 Povolené přechody: `NEW` -> `IN_PROGRESS` -> `CLOSED` (+ klidně i zpět `IN_PROGRESS` -> `NEW`, pokud chcete, ale ideálně explicitně zakázat nebo auditovat)

5) E-mailové notifikace (povinné)

Po vytvoření objednávky:

1. **Médium:** email s informací o objednávce + souhrn (nabídka, termín, ce MediaMarket-procesy a workflow poznámka).
 MediaMarket-procesy a workflow
2. **Agentura:** email potvrzení + souhrn + informac MediaMarket-procesy a workflow ntaktovat".
 MediaMarket-procesy a workflow

Doporučení:

- posílat i "status changed" e-mail agentuře při změně `NEW/IN_PROGRESS/CLOSED` (užitečné a levné).
- logovat odeslání (tabulka `EmailLog`) kvůli dohledatelnosti.

Provider: SendGrid / Mailgun / Resend / AWS SES (vybere dev; důležité je mít šablony a spolehlivost).

6) Admin dashboard – požadavky

6.1 Přehled objednávek

- list všech objednávek + stav
- filtry:
 - podle agentury
 - podle média
 - podle období (created_at)
 - podle stavu

6.2 Provize + export do fakturační šablony

- admin vybere 1+ objednávek (nebo filtr + "vybrat vše")
- zvolí
MediaMarket-procesy a workflow
nebo 5 %
MediaMarket-procesy a workflow
- systém spočítá provizi per order + součet
- export:
 - minimálně **CSV/XLSX** (pro fakturační systém / šablonu)
 - volitelně: **PDF** (pokud chcete "hotovou fakturu", ale to už je vyšší liga)

Obsah exportu (min.):

- údaje odběratele (kdo fakturuje komu – potřebujete upřesnit obchodní model; pokud zatím ne, export nechť obsahuje obě strany + identifikátory)
- seznam objednávek: `order_number`, datum, médium, agentura,
`total_pr:contentReference[oaicite:8]{index=8}ion_amount`
MediaMarket-procesy a workflow

7) GDPR / právní stránky – co backend musí hlídat

- při registraci uložit timestamp souhlasu se zpracováním osobních údajů
- při odeslání objednávky uložit timestamp souhlasu s obchodními podmínkami (doporučeno mít v objednávce pole `terms_accepted_at`)
- verzování: ideál je ukládat i `terms_version` / `privacy_version` (ať za 2 roky víš, s čím kdo souhlasil)

8) Ne-funkční požadavky (ať to nevyhoří první pondělí)

- **Bezpečnost:** RBAC + ideálně RLS na DB (pokud Supabase) / ACL v API.

- **Validace:** backend přeypočítává cenu, nedůvěruje frontendu (jinak si někdo “objedná” za 1 Kč).
 - **Rate limiting:** registrace/forgot-password/objednávky (spam).
 - **Audit log:** změny stavu objednávky a kdo to udělal.
 - **Backups:** automatické zálohy DB.
 - **Observability:** logy chyb, základní metriky.
-

9) Doporučená implementace (silný názor)

Pokud je to Lovable/Next stack, nejrychlejší a nejlevnější bývá:

- **Supabase** (Auth + Postgres + Storage + Edge Functions pro e-maily)
 - RLS pro “kdo co vidí”
 - triggers/functions pro `order_number` a email notifikace

Alternativa: vlastní API (NestJS/Express) + Postgres (Prisma) + externí auth. Je to flexibilní, ale pomalejší na dodání a dražší na údržbu.

10) Akceptační kritéria (test scénáře)

1. Agentura i médium se zaregistrují → přijde ověřovací email → po ověření se lze přihlásit.
 2. Médium vytvoří nabídku s **Cena za ks** → agentura objedná → backend spočítá **Cena celkem** správně.
 3. Médium vytvoří nabídku s **CPT** → agentura objedná s **počet zobrazení** → backend spočítá správně.
 4. Pokud nabídka má **Nutné uvést finálního klienta** → bez vyplnění nejde objednávku odeslat.
 5. Médium změní stav objednávky → agentura vidí nový stav v dashboardu (+ volitelně email).
 6. Admin vyfiltruje objednávky, vybere sazbu 2,5 % / 5 % → export sedí (součty i položky).
-

11) Co je v popisu nejspíš “chybějící minimum”

Tohle bych dal do scope jako povinné, i když to není explicitně vypsané:

- přihlášení/odhlášení + reset hesla
- CRUD nabídky (minimálně edit/archivace)
- server-side výpočty ceny + snapshot ceny v objednávce

- audit změn stavu
- log souhlasů (GDPR/terms)