

DFDS: Kompletna Tehnička Dokumentacija

Verzija: 2.0.0

Datum: Veljača 2026

Status: Producjska Verzija

Tim: Team Cloudzz

Sadržaj

- Uvod i Pregled Projekta
- Korisničke Uloge
- Tehnološki Stack
- Arhitektura Sustava
- Dashboard Funkcionalnosti
- AI Asistent
- Shema Baze Podataka
- API Dokumentacija
- Sigurnost
- Kako Pokrenuti

Uvod i Pregled Projekta

Što je DFDS?

DFDS (Developers, Founders, Deal-makers, Startups) je sveobuhvatna platforma dizajnirana da revolucionira startup ekosustav. Naš tim, Team Cloudzz, fokusirao se na rješavanje ključnog problema: *nepovezanosti između inovatora (Foundera), graditelja (Developera) i investitora.*

Misija

Stvoriti živi ekosustav gdje se:

- Startup timovi grade od nule

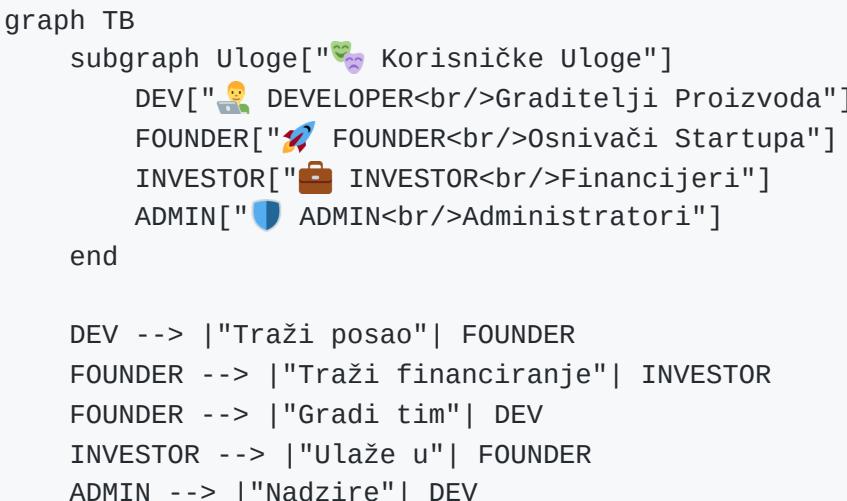
- 💰 Inovacije **financiraju** transparentno
- 🤝 Talenti **povezuju** inteligentno

Ključne Značajke Platforme

Značajka	Opis
Smart Matching	Algoritamsko povezivanje investitora i startupa korištenjem vektorske sličnosti
Real-time Chat	Privatne poruke bez kašnjenja putem WebSocketa
AI Konzultant	Instant analiza poslovnih ideja i generiranje e-mailova
Equity Simulator	Interaktivni kalkulator dilucije vlasništva
Investicijski Dashboard	Transparentno praćenje financiranja i transakcija
Community Threads	Forum za razmjenu znanja i partnerstava
API Pristup	Programatski pristup podacima platforme

👤 Korisničke Uloge

DFDS podržava četiri različite korisničke uloge, svaka s prilagođenim sučeljem i funkcionalnostima:



```
ADMIN --> |"Nadzire"| FOUNDER  
ADMIN --> |"Nadzire"| INVESTOR
```

Developer (Razvojni Programer)

Pristup: Dashboard s fokusom na mrežu i prilike

Mogućnost	Opis
Profil vještina	Prikaz tehnologija (React, Node.js, Python, itd.)
Portfolio projekata	Galerija prošlih radova s GitHub linkovima
Network pretraga	Pronalaženje drugih developera i foundera
Slanje poruka	Direktna komunikacija s potencijalnim poslodavcima
Prijava na startup	Mogućnost pridruživanja postojećim startupima

Founder (Osnivač)

Pristup: Posebni founder dashboard s metrikama startupa

Mogućnost	Opis
Kreiranje startupa	Registracija novog startupa s pitch-om i detaljima
Tim management	Pozivanje članova i dodjela uloga
Investor matching	AI-predloženi investitori na temelju fokusa
Funding runway	Prikaz prikupljenih sredstava i trajanja
Growth analytics	Grafikoni rasta i aktivnosti

Investor

Pristup: Dashboard s fokusom na deal flow

Mogućnost	Opis
Startup discovery	Pregled startupa po fazama (Pre-seed, Seed, Series A)
Portfolio praćenje	Pregled svih ulaganja

Due diligence	Pristup detaljima startupa
Direktne poruke	Kontaktiranje osnivača
Check size profil	Prikaz prosječnog ulaganja i fokusa

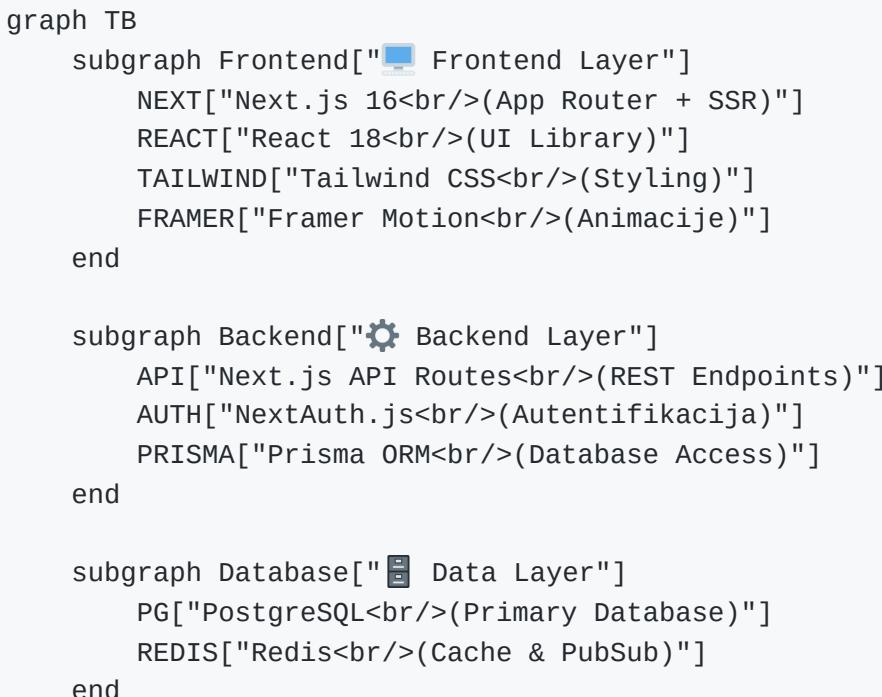
🛡 Admin

Pristup: Potpuni pristup + admin funkcije

Mogućnost	Opis
Analytics dashboard	Metrike platforme u realnom vremenu
Blog upravljanje	Kreiranje i uređivanje blog postova
Korisnici pregled	Nadzor svih računa
Feature requests	Upravljanje zahtjevima za nove funkcije

🔧 Tehnološki Stack

Pregled Arhitekture



```

subgraph Services ["cloud External & Local Services"]
    SOKETI["Soketi<br/>(Local WebSocket Server)"]
    RESEND["Resend<br/>(Transactional Email)"]
    OPENAI["OpenAI<br/>(AI Assistant)"]
    POSTHOG["PostHog<br/>(Analytics)"]
end

NEXT --> API
REACT --> NEXT
TAILWIND --> REACT
FRAMER --> REACT

API --> AUTH
API --> PRISMA
PRISMA --> PG
API --> REDIS

API --> SOKETI
API --> RESEND
API --> OPENAI
NEXT --> POSTHOG

```

Tehnologija u Službi Korisnika (UX/DX)

Naš odabir tehnologija nije slučajan. Svaki alat odabran je s ciljem da maksimizira performanse, sigurnost i fluidnost korištenja. Ovako se naš stack prevodi u konkretnе benefite za korisnika:

Frontend: Brzina i Fluidnost

Korisničko sučelje je prvo što korisnik vidi. Koristimo najmodernije alate kako bi ono bilo **trenutačno, responzivno i ugodno**.

Tehnologija	Uloga u Sustavu	 UX Benefit za Korisnika
Next.js 16	Core Framework. Upravlja rutiranjem, renderiranjem (SSR/RSC) i optimizacijom.	Instantno učitavanje. Stranice se generiraju na serveru, pa korisnik ne gleda "loading" spinneve već odmah vidi sadržaj.
React 18	UI Library. Omogućuje modularnu izgradnju sučelja.	Interaktivnost. Aplikacija reagira na klikove bez osvježavanja cijele stranice (SPA osjećaj).

Tailwind CSS	Styling Engine. Utility-first CSS za dizajn.	Konzistentnost. Sučelje izgleda savršeno na svim uređajima, od mobitela do 4K monitora. Dark mode je ugrađen i ugodan za oči.
Framer Motion	Animation Library. Upravlja tranzicijama.	Prirodan osjećaj. Elementi ne "iskaču" naglo već fluidno klize, dajući osjećaj visoke kvalitete i poliranosti.
Lucide React	Icon System. Vektorske ikone.	Jasnoća. Lagane i prepoznatljive ikone pomažu korisnicima da intuitivno navigiraju sučeljem.

⚙️ Backend: Stabilnost i Real-time Komunikacija

Backend je nevidljivi motor koji osigurava da podaci budu točni, sigurni i dostavljeni na vrijeme.

Tehnologija	Uloga u Sustavu	💡 UX Benefit za Korisnika
Next.js API	Serverless API. Backend logika integrirana s frontendom.	Brzi odziv. API pozivi su optimizirani i nalaze se "blizu" frontenda, smanjujući latenciju.
Soketi	WebSocket Server. Lokalna, brza alternativa Pusheru.	Real-time Chat. Poruke i notifikacije stižu odmah dok ste online, bez potrebe za osvježavanjem stranice.
Prisma ORM	Database Layer. Sigurna komunikacija s bazom.	Pouzdanost. Smanjuje greške u podacima, osiguravajući da su vaše transakcije i poruke uvijek točno spremljene.
Redis	Caching & Pub/Sub. Privremena memorija.	Brzina. Često korišteni podaci se učitavaju iz memorije umjesto diska, čineći aplikaciju "snappy".
PostgreSQL	Baza Podataka. Glavno spremište podataka.	Sigurnost podataka. Vaši podaci su strukturirani, sigurni i konzistentni (ACID compliance).

AI & Integracije: Pametna Pomoć

Napredne značajke koje štede vrijeme i daju vrijednost.

Tehnologija	Uloga u Sustavu	UX Benefit za Korisnika
OpenAI (GPT-4)	AI Intelligence. Pokreće "AI Asistenta".	Pametni savjeti. Korisnik dobiva trenutnu analizu pitcha ili draft emaila, štedeći sate rada na pisanju.
Resend	Email Service. Transakcijski emailovi.	Pouzdane obavijesti. Važne informacije (reset lozinke, pozivnice) ne završavaju u spamu.
PostHog	Analytics. Praćenje korištenja (anonimizirano).	Stalna poboljšanja. Analiziramo kako koristite aplikaciju da bismo popravili bugove i poboljšali UX prije nego ih i primijetite.

Arhitektura Sustava

Kako Podaci Putuju

```
sequenceDiagram
    participant U as  Korisnik
    participant F as  Frontend (Next.js)
    participant A as  API Routes
    participant P as  Prisma
    participant DB as  PostgreSQL
    participant R as  Redis
    participant WS as  Pusher
    participant AI as  OpenAI

    U->>F: Klik na "Pošalji poruku"
    F->>A: POST /api/messages/send
    A->>P: prisma.message.create()
    P->>DB: INSERT INTO Message
    DB-->>P: OK + message data
    P-->>A: Message object
    A->>R: PUBLISH channel:user_123
    R->>WS: Trigger event
    WS-->>U: Real-time poruka stiže!
    A-->>F: 201 Created
```

F-->U: UI Update ✓

Note over U,AI: AI Asistent Flow
U->F: "Analiziraj moj pitch"
F->A: POST /api/chat
A->AI: OpenAI Completion
AI-->A: AI Response
A-->F: Stream response
F-->U: Markdown renderiran odgovor

Struktura Direktorija (Sažetak)

Napomena: Ovo je sažeti prikaz glavnih direktorija. Stvarna struktura sadrži više datoteka.

```
dfds/
├── app/                      # Next.js App Router
│   ├── api/                  # 34+ API endpoints
│   │   ├── auth/             # NextAuth handlers
│   │   ├── messages/         # Chat API
│   │   ├── startups/          # Startup CRUD
│   │   ├── threads/           # Forum API
│   │   ├── transactions/      # Payment API
│   │   └── v1/                # Public API v1
│   ├── dashboard/            # 15+ dashboard stranica
│   │   ├── page.tsx          # Glavni dashboard
│   │   ├── startups/          # Startup pregled
│   │   ├── network/           # Developer mreža
│   │   ├── investors/          # Investitori
│   │   ├── messages/           # Chat sučelje
│   │   ├── payments/           # Transakcije
│   │   ├── threads/            # Community forum
│   │   ├── roadmap/            # Feature voting
│   │   ├── calculator/         # Equity simulator
│   │   ├── api-access/          # API ključevi
│   │   ├── members/             # Članstvo
│   │   └── settings/            # Postavke profila
│   └── (public pages)/        # Landing, Login, Register...
├── components/                # React komponente
│   ├── ui/                   # Shadcn-style UI primitivi
│   ├── dashboard/             # Dashboard komponente
│   ├── landing/               # Landing page sekcije
│   ├── messaging/              # Chat komponente
│   ├── payments/                # Payment komponente
│   └── ai/                    # AI Asistent
└── lib/                      # Utility funkcije
```

```
|   ├── auth.ts          # NextAuth konfiguracija
|   ├── prisma.ts        # Prisma client
|   ├── pusher.ts         # WebSocket setup
|   ├── matchmaker.ts    # Vector similarity matching
|   └── email.ts         # Email templates
|
├── prisma/
|   ├── schema.prisma   # 25+ modela
|   └── seed.ts          # Test podaci
└── config/
    └── nav.ts           # Navigacija
```

Struktura Frontend Aplikacije

Aplikacija je izgrađena na **Next.js App Routeru** (verzija 16.x) i koristi hijerarhijsku strukturu ruta.

Mapa Ruta

```
graph TD
Root["/ (Root Layout)"]

%% Public Routes
Root --> Landing["/ (Landing Page)"]
Root --> About["/about"]
Root --> Contact["/contact"]
Root --> Docs["/docs"]
Root --> Blog["/blog"]
Blog --> BlogPost["/blog/[slug]"]

%% Auth Routes
Root --> Login["/login"]
Root --> Join["/join"]
Root --> Verify["/verify-email"]
Root --> Reset["/reset-password"]

%% Discovery
Root --> Discover["/discover"]
Root --> PublicProfile["/profile/[id]"]

%% Dashboard (Protected)
Root --> DashLayout["/dashboard (Layout)"]
DashLayout --> DashOverview["/dashboard (Overview)"]
DashLayout --> DashNetwork["/dashboard/network"]
```

```

DashLayout --> DashStartups["/dashboard/startups"]
DashLayout --> DashInvestors["/dashboard/investors"]
DashLayout --> DashMessages["/dashboard/messages"]
DashLayout --> DashSettings["/dashboard/settings"]

%% Admin/Creator
Root --> Creator["/creator/dashboard"]
Root --> BlogAdmin["/blog/admin"]

```

Detaljni Opis Stranica

1. Javne Stranice (Public Pages)

Stranice dostupne svim posjetiteljima bez prijave.

Ruta	Opis	Ključne Komponente
app/page.tsx	Naslovna stranica. Prikazuje "Hero" sekciju, vizualizaciju mreže ("NetworkPulse") i CTA gumbu.	<HeroContent> , <NetworkPulse> , <DynamicIsland>
app/about/page.tsx	O nama. Misija tima, vizija projekta i informacije o "Team Cloudzz".	<GlassCard> , <Section>
app/discover/page.tsx	Discover. Centralno mjesto za pretragu startupa, developera i investitora bez registracije (read-only pregled).	<Tabs> , <Input> (Search), <Avatar>
app/blog/page.tsx	Blog. Lista članaka s novostima i uvidima. Koristi prisma za dohvatanje objavljenih postova.	<GlassCard> , date-fns
app/blog/[slug]/page.tsx	Blog Post. Pojedinačni članak. Renderira Markdown sadržaj sigurno.	react-markdown , remark-gfm

<code>app/contact/page.tsx</code>	Kontakt. Forma za kontakt i linkovi na društvene mreže.	<code>mailto:</code> , PostHog tracking
<code>app/docs/page.tsx</code>	Dokumentacija. Renderira <code>DFDS_Dokumentacija.md</code> unutar aplikacije.	<code>fs</code> , <code>path</code> , Markdown rendering

2. Autentifikacija (Auth Pages)

Stranice za upravljanje identitetom korisnika.

Ruta	Opis	Logika
<code>app/join/page.tsx</code>	Registracija. Multi-step forma za odabir uloge (Developer/Founder/Investor) i unos podataka.	POST <code>/api/register</code> , PostHog event <code>user_signed_up</code>
<code>app/login/page.tsx</code>	Prijava. Prijava emailom/lozinkom i "Demo Account" opcija.	<code>signIn("credentials")</code> , Session update
<code>app/verify-email/page.tsx</code>	Verifikacija. Unos 6-znamenkastog koda poslanog na email.	POST <code>/api/verify-email</code> , Auto-redirect
<code>app/reset-password/page.tsx</code>	Reset Lozinke. Zahtjev za reset i postavljanje nove lozinke.	Token validacija, Secure hash update

3. Dashboard (Protected Pages)

Zaštićeni dio aplikacije unutar `app/dashboard/layout.tsx`.

Layout: Sadrži bočnu traku (`<Sidebar>`), gornju traku s pretragom (`<DashboardSearch>`) i `<UserNav>`. Provjerava sesiju i uloge.

Ruta	Funkcionalnost	Tehnički Detalji
<code>dashboard/page.tsx</code>	Pregled. Prikazuje različite widgete ovisno o ulozi	<code>Promise.all</code> paralelni dohvati statistike, rasta i aktivnosti.

	(Founder vs Developer).	
dashboard/messages/page.tsx	Poruke. Chat sustav. Podržava desktop i mobile view (responzivno).	<code>useSearchParams</code> za selekciju konverzacije, Real-time updates.
dashboard/network/page.tsx	Mreža. Grid prikaz developer-a s "Infinite Scroll" logikom.	Server-side initial fetch + Client-side load more.
dashboard/startups/page.tsx	Startipi. Pregled startup-a s filtriranjem i paginacijom.	<code>prisma.startup.findMany</code> s filterima.
dashboard/settings/page.tsx	Postavke. Upravljanje profilom, notifikacijama i brisanjem računa (Soft Delete).	Tab navigacija, Optimistic UI updates.
dashboard/api-access/page.tsx	Developer API. Generiranje i upravljanje API ključevima.	Prikaz ključa samo jednom (sigurnost), Copy-to-clipboard.

4. Creator & Admin Zone

Posebne stranice za administratore i kreatore sadržaja.

Ruta	Opis	Kontrola Pristupa
app/creator/dashboard	Analitika. Prikaz KPI-jeva sustava, grafova rasta i telemetrije.	Provjera <code>role === 'ADMIN'</code> .

app/blog/admin	CMS. Sučelje za pisanje, uređivanje i objavu blog postova.	Provjera role ==== 'ADMIN' .
----------------	---	---------------------------------

🎨 Globalne Komponente i Layout

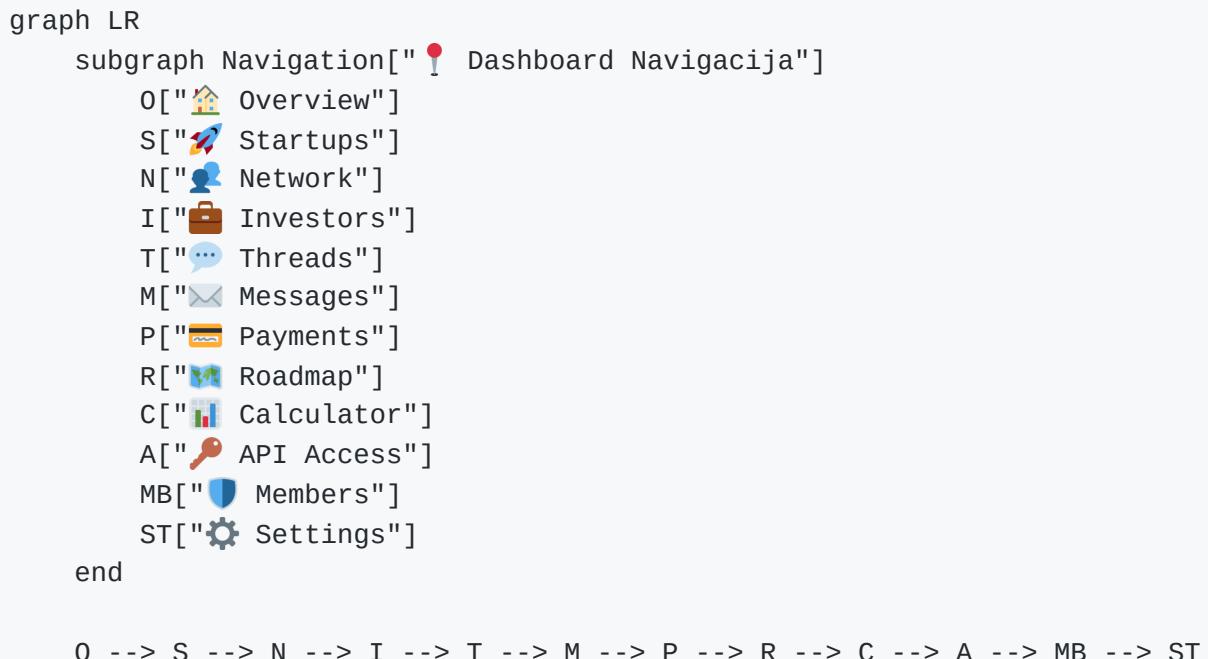
Datoteka `app/layout.tsx` je korijen aplikacije i definira globalno stanje:

- **Providers:** `AuthProvider` (NextAuth), `PostHogProvider` (Analitika), `ThemeProvider` (Dark mode).
- **UI Elementi:** `DynamicBackground` (Animirana pozadina), `CommandMenu` (CMD+K izbornik), `Toaster` (Notifikacije).
- **Metadata:** SEO konfiguracija, OpenGraph slike.

Datoteka `app/globals.css` definira Tailwind CSS varijable, animacije (shimmer, pulse) i "glassmorphism" uslužne klase.

📱 Dashboard Funkcionalnosti (Sve Kartice)

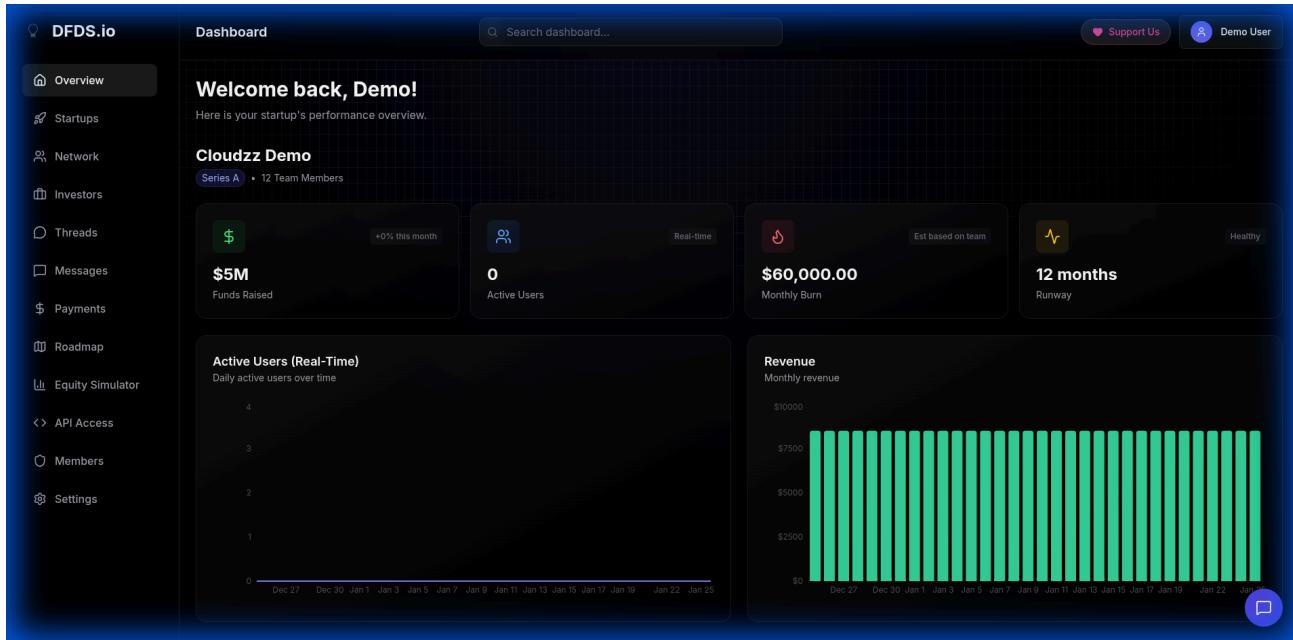
Dashboard je srce DFDS platforme. Sadrži **12 zasebnih kartica** za različite funkcionalnosti:



1. Overview (Pregled)

Ruta: /dashboard

Kontrolna ploča s pregledom ključnih metrika i brzim akcijama.



Komponente:

Element	Opis
Welcome Section	Personalizirani pozdrav s imenom korisnika
Stats Cards	4 kartice: Connections, Startups, Investors, Growth
Growth Dashboard	Interaktivni graf s metrikama rasta (za developere)
Founder Dashboard	Runway, funding, team size (za foundere)
Quick Actions	Brzi linkovi: Find Co-founders, Browse Startups, Connect with Investors
Action Buttons	Messages i Payments gumbi s gradijentom
Recent Activity	Lista zadnjih aktivnosti s ikonama i vremenskim oznakama

Razlike po ulozi:

```

graph TB
    subgraph Dashboard
        CHECK{Uloga?}
        CHECK -->|FOUNDER| FD["FounderDashboard<br/>• Runway<br/>• Raised<br/>• Connectior"]
        CHECK -->|DEVELOPER/INVESTOR| GD["GrowthDashboard<br/>• Connectior"]
    end

```

2. 🚀 Startups

Ruta: /dashboard/startups

Pregled i kreiranje startupa na platformi.

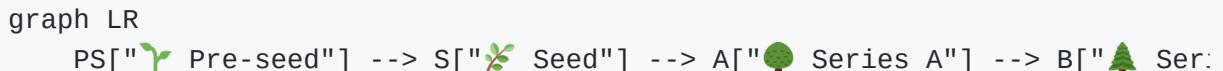
Funkcionalnosti:

Funkcija	Opis
Search	Pretraga po imenu, pitch-u, fazi ili founderu
Create Startup	Modal za registraciju novog startupa
Startup Cards	Prikaz: logo, ime, faza, pitch, founder
Connect Button	Direktno slanje poruke founderu
Pagination	Navigacija kroz stranice (25 po stranici)

External Links

Link na web stranicu startupa

Faze Startupa:



3. 💬 Network (Mreža Developera)

Ruta: /dashboard/network

Pronalaženje talenata za gradnju tima.

The screenshot shows the DFDS.io developer network interface. On the left is a sidebar with navigation links: Overview, Startups, Network (selected), Investors, Threads, Messages, Payments, Roadmap, Equity Simulator, API Access, Members, and Settings. The main area is titled 'Developer Network' and has a subtitle 'Find talented developers to join your team'. It features a grid of six developer profiles, each with a circular icon, name, title, location, skills, and a 'Send Message' button. The profiles are: Betty Gonzalez (Developer, Toronto, CA, React, Rust, Docker, PostgreSQL), Sarah Jones (Developer, London, UK, Next.js, Redis), David Wilson (Developer, New York, NY, MongoDB, Next.js, AWS), Robert Martin (Developer, Berlin, DE, Svelte, PostgreSQL, Python), Mark Hernandez (Developer, Austin, TX, Next.js, MongoDB, Docker), and Michael Lopez (Developer, Boston, MA, React, GraphQL). There are also search bars at the top and bottom right.

Funkcionalnosti:

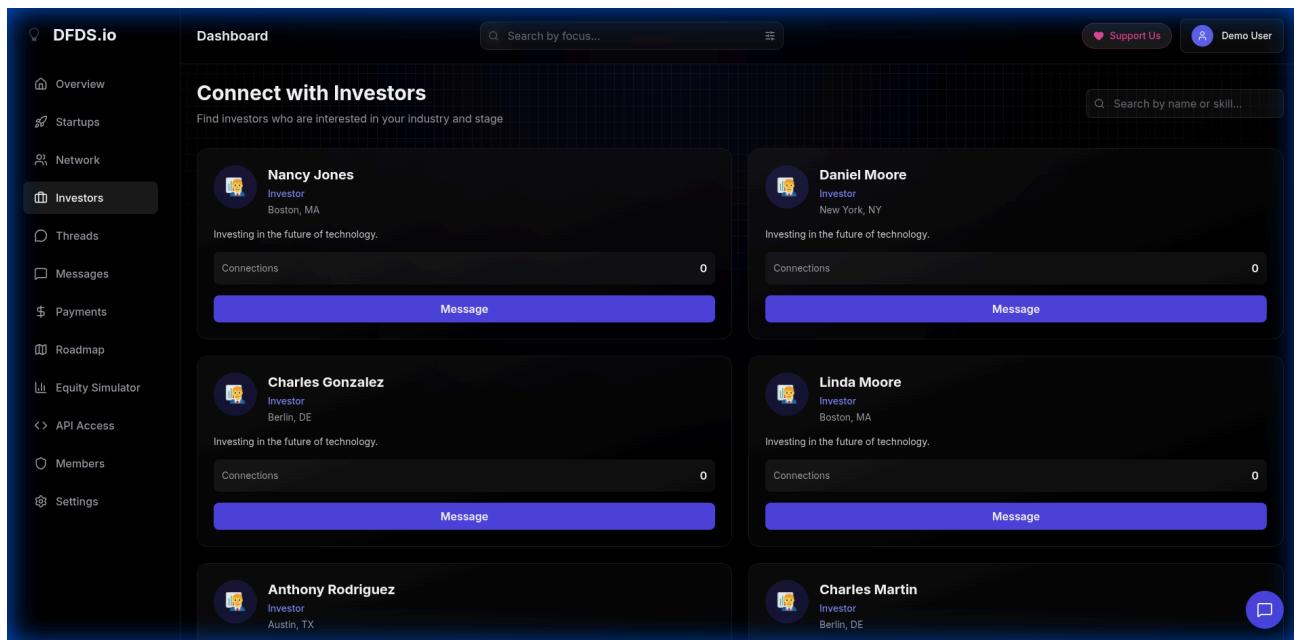
Funkcija	Opis
Developer Grid	Kartice developera s avatarima
Search	Pretraga po imenu ili emailu
Skills Display	Prikaz tehnologija (badges)
Bio Preview	Kratki opis developera

Location	Geografska lokacija
Connect	Gumb za slanje poruke
Load More	Infinite scroll ili pagination

4. Investors

Ruta: /dashboard/investors

Povezivanje s investitorima.



The screenshot shows the DFDS.io dashboard with a dark theme. On the left is a sidebar with navigation links: Overview, Startups, Network, Investors (selected), Threads, Messages, Payments, Roadmap, Equity Simulator, API Access, Members, and Settings. The main area is titled 'Connect with Investors' and contains a search bar. Below it, a sub-header says 'Find investors who are interested in your industry and stage'. Six investor profiles are listed in a grid:

- Nancy Jones**, Investor, Boston, MA. Message button.
- Daniel Moore**, Investor, New York, NY. Message button.
- Charles Gonzalez**, Investor, Berlin, DE. Message button.
- Linda Moore**, Investor, Boston, MA. Message button.
- Anthony Rodriguez**, Investor, Austin, TX. Message button.
- Charles Martin**, Investor, Berlin, DE. Message button.

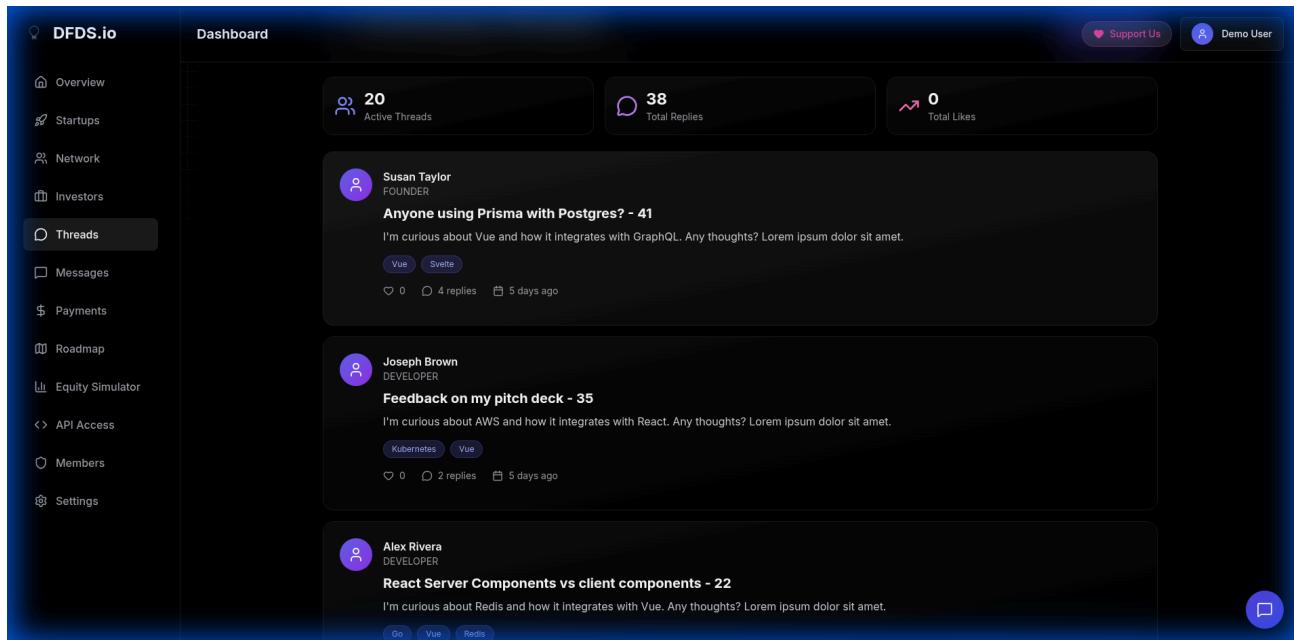
Funkcionalnosti:

Funkcija	Opis
Investor Grid	Kartice investitora
Search	Pretraga po imenu
Focus Areas	Područje interesa (SaaS, AI, Web3...)
Check Size	Prosječna veličina ulaganja
Portfolio Count	Broj ulaganja
Connect	Direktna poruka investitoru

5. Threads (Diskusije)

Ruta: /dashboard/threads

Community forum za razmjenu znanja.



Funkcionalnosti:

Funkcija	Opis
Create Thread	Kreiranje nove diskusije s naslovom, sadržajem i tagovima
Thread List	Lista svih diskusija sortirano po vremenu
Tags	Filtriranje po kategorijama
Like	Heart reakcija na thread
Reply	Odgovaranje na diskusije
Author Info	Ime, uloga i firma autora
Trending	Sidebar s popularnim threadovima

Thread Model:

```
erDiagram
    Thread ||--o{ ThreadReply : "ima"
    Thread ||--o{ ThreadLike : "ima"
    User ||--o{ Thread : "kreira"
```

```

User ||--o{ ThreadReply : "piše"
User ||--o{ ThreadLike : "lajka"

Thread {
    string id PK
    string title
    string content
    string[] tags
    datetime createdAt
}

ThreadReply {
    string id PK
    string content
    int likes
    datetime createdAt
}

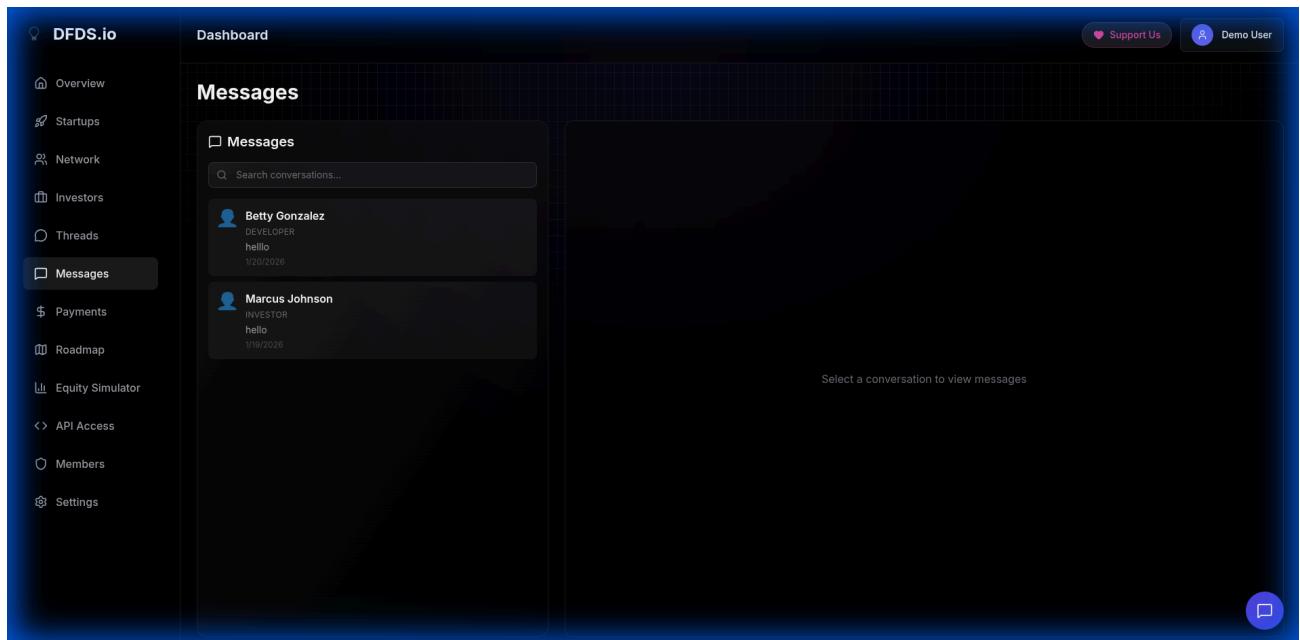
ThreadLike {
    string id PK
    datetime createdAt
}

```

6. Messages (Poruke)

Ruta: /dashboard/messages

Real-time chat sustav.



The screenshot shows the DFDS.io dashboard with the 'Messages' section selected in the sidebar. The main area displays two conversations:

- Betty Gonzalez (Developer)**: hello, 1/20/2026
- Marcus Johnson (Investor)**: hello, 1/19/2026

A message at the bottom right says "Select a conversation to view messages".

Funkcionalnosti:

Funkcija	Opis
Message Inbox	Lista konverzacija s preview-om zadnje poruke
Conversation View	Prikaz svih poruka u konverzaciji
Real-time Updates	Pusher WebSocket za instant poruke
Unread Indicators	Označavanje nepročitanih poruka
User Search	Pretraga korisnika za novu konverzaciju
Timestamps	Relativno vrijeme (prije X minuta)

Real-time Arhitektura:

```
sequenceDiagram
    participant A as Alice
    participant API as Server
    participant P as Pusher
    participant B as Bob

    A->>API: POST /api/messages/send
    API->>API: Spremi u PostgreSQL
    API->>P: pusher.trigger('user-bob', 'new-message', data)
    P-->>B: WebSocket event
    B->>B: Prikaži novu poruku
    API-->>A: 201 OK
```

7. Payments (Plaćanja)

Ruta: `/dashboard/payments`

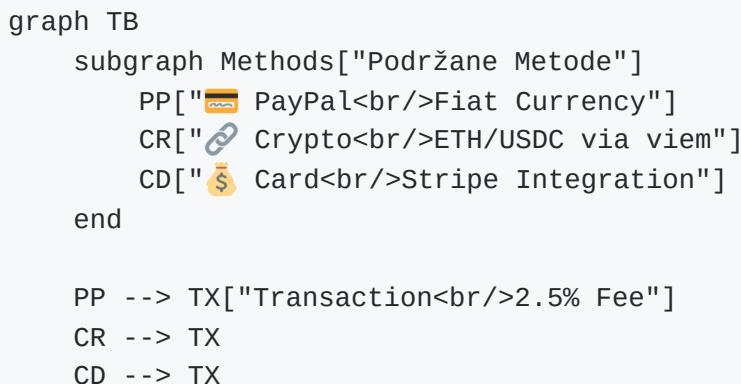
Slanje i primanje novca unutar platforme.

Funkcionalnosti:

Funkcija	Opis
Stats Overview	Total Sent, Total Received, Platform Fees (2.5%)

User Search	Pretraga primatelja po imenu ili emailu
Payment Modal	Modal za unos iznosa i odabir metode
Transaction History	Lista svih transakcija s filterima
Payment Methods	PayPal, Crypto (vjem), Card

Platne Metode:



Transaction Statusi:

Status	Opis
PENDING	Transakcija pokrenuta, čeka potvrdu
COMPLETED	Uspješno izvršena
FAILED	Neuspješna (nedovoljno sredstava, etc.)
CANCELLED	Otkazana od strane korisnika

8. Roadmap

Ruta: </dashboard/roadmap>

Transparentni prikaz planiranih funkcionalnosti s mogućnošću glasanja.

Funkcionalnosti:

Funkcija	Opis
Feature Requests	Lista predloženih funkcionalnosti
Voting	Upvote/Downvote sustav
Status Tracking	PLANNED → IN_PROGRESS → COMPLETED
Submit Request	Predlaganje novih funkcionalnosti
Sort Options	Po glasovima, datumu, statusu

9. **Equity Simulator (Kalkulator)**

Ruta: /dashboard/calculator

Interaktivni alat za vizualizaciju dilucije vlasništva kroz runde financiranja.

Funkcionalnosti:

Funkcija	Opis
Initial Equity	Postavljanje početnih postotaka (Founder, Co-founder, Investors)
Add Funding Round	Simulacija Pre-seed, Seed, Series A...

Dilution Visualization	Pie chart s promjenama postotaka
Scenario Comparison	Usporedba različitih scenarija
Export	Dijeljenje rezultata

Primjer Dilucije:

```
pie title "Prije Seed Runde"
  "Founder" : 60
  "Co-founder" : 30
  "Angel" : 10
```

```
pie title "Nakon Seed Runde ($1M @ $5M valuation)"
  "Founder" : 48
  "Co-founder" : 24
  "Angel" : 8
  "Seed Investors" : 20
```

10. API Access

Ruta: </dashboard/api-access>

Upravljanje API ključevima za programatski pristup.

Funkcionalnosti:

Funkcija	Opis
API Key Generation	Kreiranje novih ključeva s imenima
Key Management	Lista svih ključeva s statusom
Permissions	Odabir dozvola (read, write, delete)
Expiration	Postavljanje isteka ključa
Documentation	Inline API dokumentacija s primjerima

Test Endpoint	Testiranje API poziva iz UI-a
Usage Stats	Prikaz zadnjeg korištenja

API Key Model:

```

erDiagram
    User ||--o{ ApiKey : "posjeduje"

    ApiKey {
        string id PK
        string keyHash "SHA-256 hash"
        string keyPrefix "Prvih 8 znakova"
        string name
        boolean isActive
        string[] permissions
        datetime lastUsed
        datetime expiresAt
    }

```

11. Members

Ruta: [/dashboard/members](#)

Prikaz verificiranih članova i premium pristupa.

Funkcionalnosti:

Funkcija	Opis
Verified Builders	Lista verificiranih developera
Subscription Tiers	FREE, PRO, GROWTH planovi
Badge Display	Verifikacijski badge za profile

Subscription Tiers:

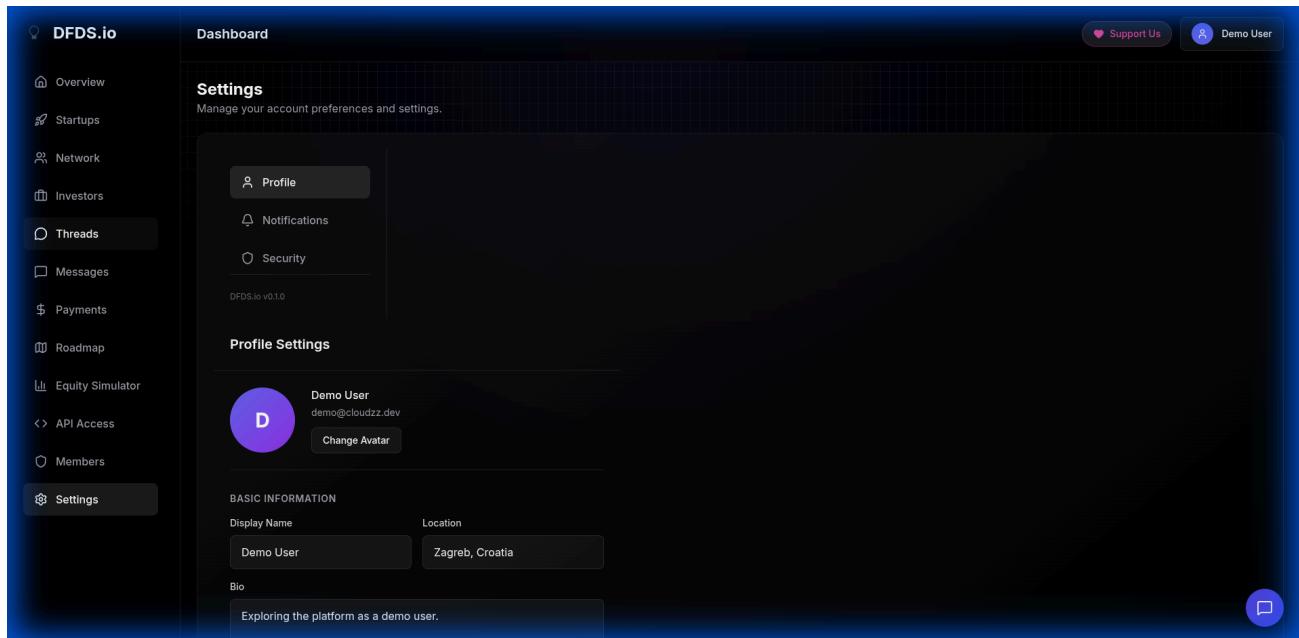
Tier	Mogućnosti
FREE	Osnovni pristup, 100 poruka/dan

PRO	Sve FREE + neograničene poruke, API pristup
GROWTH	Sve PRO + prioritetna podrška, advanced analytics

12. Settings (Postavke)

Ruta: /dashboard/settings

Upravljanje korisničkim računom i profilom.



Tab: Profile

Polje	Opis
Name	Ime i prezime
Email	Email adresa (readonly)
Bio	Kratki opis
Location	Grad/Država
Skills	Lista vještina (tags)
GitHub URL	Link na GitHub profil
LinkedIn URL	Link na LinkedIn profil

Tab: Account

Akcija	Opis
Change Password	Ažuriranje lozinke
Delete Account	Brisanje računa (30-dnevni grace period)

Tab: Notifications

Opcija	Opis
Email Notifications	Toggle za email obavijesti
Push Notifications	Toggle za push obavijesti
Marketing Emails	Toggle za marketing

AI Asistent

Floating widget dostupan na svim dashboard stranicama.

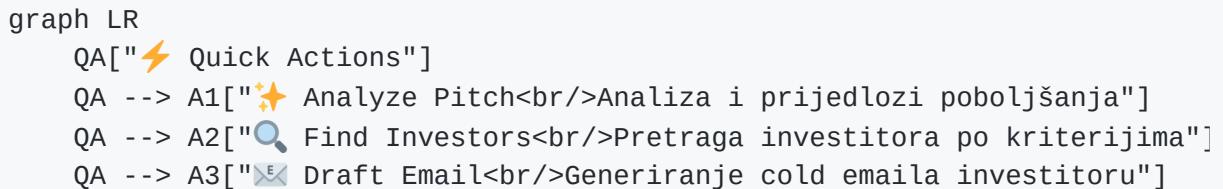
Lokacija u Kodu

/components/ai/AiAssistant.tsx

Funkcionalnosti

Funkcija	Opis
Chat Interface	Razgovor s AI asistentom
Quick Actions	Unaprijed definirane akcije
Resizable Window	Povlačenje za promjenu veličine
Markdown Rendering	Formatirani odgovori s code highlightingom
Conversation History	Pamćenje konteksta razgovora

Quick Actions



Primjer Korištenja

User: "Analiziraj moj pitch: Gradimo AI platformu za automatizirano testiranje novih ideja." AI: ### Analiza Pitcha

Snage:

- Jasno definirani problem
- Rastući TAM (Total Addressable Market)

Prijedlozi:

1. Dodajte konkretnе brojke (smanjenje vremena testiranja za X%)
2. Navedite konkureniju i vašu diferencijaciju
3. Uključite social proof (beta korisnici, partnerships)

Ocjena: 7/10 - Dobar temelj, treba više specifičnosti.

☰ Shema Baze Podataka i ORM

Projekt koristi **Prisma ORM** za upravljanje PostgreSQL bazom. Shema je definirana u `prisma/schema.prisma`.

Entity Relationship Diagram

```

erDiagram
    User ||--o| Profile : "ima"
    User ||--o{ Startup : "osniva"
    User ||--o{ StartupMembership : "član"
    User ||--o{ Message : "šalje"
    User ||--o{ Connection : "follower"
    User ||--o{ Connection : "following"
    User ||--o{ Thread : "kreira"
    User ||--o{ Transaction : "sender"

```

```
User ||--o{ Transaction : "receiver"
User ||--o{ ApiKey : "posjeduje"
User ||--o{ Notification : "prima"

Profile ||--o{ Project : "ima"

Startup ||--o{ StartupMembership : "članovi"
Startup ||--o| Team : "tim"

Team ||--o{ TeamMembership : "članovi"
Team ||--o{ TeamInvite : "pozivnice"

Conversation ||--o{ Message : "sadrži"
Conversation ||--o{ ConversationParticipant : "sudionici"

Thread ||--o{ ThreadReply : "odgovori"
Thread ||--o{ ThreadLike : "lajkovi"

FeatureRequest ||--o{ FeatureRequestVote : "glasovi"

User {
    string id PK
    string name
    string email UK
    datetime emailVerified
    string password
    UserRole role
    SubscriptionTier subscriptionTier
    string referralCode UK
    int referralCount
    boolean isVerifiedBuilder
    datetime deletedAt
    datetime scheduledDeletionAt
}

Profile {
    string id PK
    string bio
    string location
    string[] skills
    string githubUrl
    string linkedinUrl
    string experience
    string availability
    string rate
    string firm
    string checkSize
    string focus
}
```

```

Startup {
    string id PK
    string name
    string pitch
    string stage
    string websiteUrl
    string raised
    int teamSize
}

Transaction {
    string id PK
    float amount
    float serviceFee
    float netAmount
    PaymentProvider provider
    TransactionStatus status
    string idempotencyKey UK
}

Message {
    string id PK
    string content
    boolean read
    string[] attachments
}

```

Seeding Skripte (Početni Podaci)

Sustav sadrži napredne skripte za generiranje testnih podataka koje olakšavaju razvoj i testiranje.

Skripta	Lokacija	Opis
seed.ts	prisma/seed.ts	Glavna seed skripta. Kreira "Demo Usera", 50 developer, 10 investitora, 20 foundera, te generira početne startupe, threadove i metrike rasta.
seed-large.ts	prisma/seed-large.ts	Stress test seed. Generira masivni dataset od 1500+ korisnika (500 dev, 500 inv, 500 founder) za testiranje performansi baze i UI-a.

<code>seed-chat.ts</code>	<code>prisma/seed-chat.ts</code>	Chat simulator. Kreira realistične razgovore između demo korisnika i drugih sudionika kako bi inbox izgledao živo.
<code>verify-counts.ts</code>	<code>prisma/verify-counts.ts</code>	Pomoćna skripta za brzu provjeru broja zapisa u bazi po kategorijama.

Utility Library (@lib)

Modularni set pomoćnih funkcija koji pokreće backend logiku.

Modul	Svrha	Ključne Funkcije
<code>auth.ts</code>	Konfiguracija NextAuth.js-a.	<code>authOptions</code> (Providers: GitHub, Google, Credentials), custom <code>signIn</code> callback za soft-delete provjera.
<code>prisma.ts</code>	Instanciranje Prisma klijenta.	Singleton instanca za sprječavanje "too many connections" grešaka u dev modu.
<code>pusher.ts</code>	WebSocket konfiguracija.	Klijent i Server instance za Soketi (kompatibilan s Pusherom).
<code>api-key-auth.ts</code>	Autentifikacija API ključeva.	<code>authenticateApiKey</code> , <code>withApiKeyAuth</code> - provjera Bearer tokena i dozvola.
<code>rate-limit.ts</code>	Zaštita od zlouporabe.	Redis-backed limiteri: <code>general</code> (60/min), <code>auth</code> (10/min), <code>chat</code> (10/min).
<code>matchmaker.ts</code>	Algoritam za povezivanje.	<code>Matchmaker</code> klasa koja koristi vektorsku sličnost (simulirano) za spajanje investitora i foundera.
<code>dashboard-queries.ts</code>	Optimizirani dohvati podataka.	<code>unstable_cache</code> funkcije za brzo učitavanje dashboarda (statistika, rast, aktivnost).

<code>user-cleanup.ts</code>	GDPR compliance.	Logika za trajno brisanje "soft-deleted" korisnika nakon isteka grace perioda (30 dana).
------------------------------	------------------	--

🔗 Custom Hooks (@hooks)

React hookovi koji inkapsuliraju kompleksnu logiku na klijentu.

Hook	Opis
<code>useConversationSubscription</code>	Upravlja real-time chatom. Sluša Pusher kanale, ažurira state poruka, hendla "typing" indikatore i optimistično dodaje poruke.
<code>useDebounce</code>	Odgađa izvršavanje funkcije (npr. pretrage) dok korisnik ne prestane tipkati, smanjujući broj API poziva.

🔧 Maintenance Scripts (@scripts)

Pomoćne TypeScript i Bash skripte za održavanje i verifikaciju sustava.

Skripta	Opis
<code>smoke-test.sh</code>	Brza provjera zdravlja sustava (Healthcheck, DB connectivity, HTTP 200 na homepage).
<code>verify-user-deletion.ts</code>	Testira flow brisanja korisnika. Kreira testnog korisnika i ovisne zapise, briše ga, te potvrđuje da je "cascade delete" obrisao sve povezane podatke.
<code>verify-middleware.ts</code>	Testira sigurnosna pravila middleware-a (npr. blokiranje neautentificiranih poziva na <code>/dashboard</code>).
<code>verify-production-upgrade.ts</code>	Provjerava integritet podataka nakon upgrade-a (npr. da svaki Startup ima Team i Founder je OWNER).
<code>GenerateApiKey.ts</code>	Pomoćni alat za ručno generiranje API ključeva za testiranje.

🔌 API Dokumentacija

🛡️ Autentifikacija

API podržava dva primarna mehanizma autentifikacije:

- Session-based (Cookie)**: Koristi se za frontend aplikaciju (NextAuth.js).
 - Header: Automatski se šalje putem kolačića.
- API Key (Bearer Token)**: Koristi se za eksterne integracije i v1 endpointove.
 - Header: `Authorization: Bearer sk_live_...`

🌐 Javni API (v1)

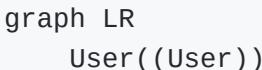
Namijenjen za eksterne integracije i automatizaciju. Zahtjeva važeći API ključ.

Metoda	Endpoint	Opis	Auth
GET	/api/v1/startups	Dohvaća listu startupa s paginacijom.	API Key
POST	/api/v1/ingest/metrics	Ingestija metrika rasta (revenue, users).	API Key
POST	/api/v1/investor/portfolio	Ažuriranje portfolio kompanija za investitore.	API Key
POST	/api/v1/startup/funding	Ažuriranje podataka o financiranju startupa.	API Key
POST	/api/v1/team-sync	Sinkronizacija članova tima iz vanjskih sustava.	API Key

🔧 Interni API Endpoints

Ovi endpointi koriste se primarno unutar web aplikacije i oslanjaju se na session autentifikaciju.

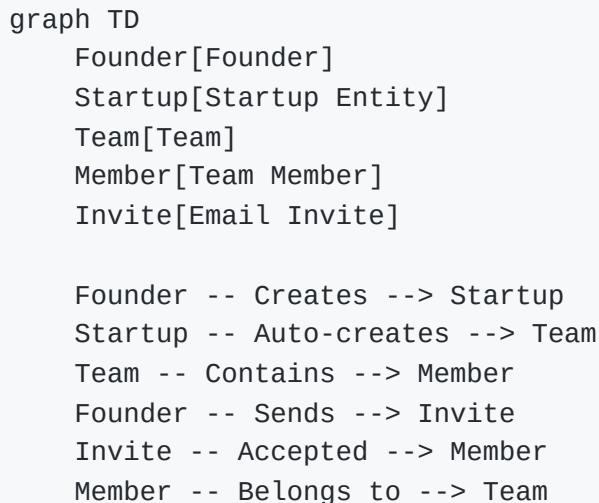
👤 Auth & Korisnici



API[API Gateway]
DB[(Database)]
Email[Resend Service]

User -- Register --> API
API -- Create User --> DB
API -- Send Email --> Email
Email -- Verification Code --> User
User -- Verify --> API
API -- Update Status --> DB
User -- Login --> API
API -- Session Cookie --> User

Metoda	Endpoint	Opis
POST	/api/register	Registracija novog korisnika.
POST	/api/verify-email	Verifikacija email adrese putem koda.
POST	/api/resend-verification	Ponovno slanje verifikacijskog emaila.
POST	/api/auth/reset-password/request	Zahtjev za reset lozinke.
POST	/api/auth/reset-password/confirm	Potvrda nove lozinke.
POST	/api/applications/join	Javna forma za prijavu interesa.
GET	/api/profiles/[id]	Dohvat javnog profila korisnika.
GET	/api/settings	Dohvat postavki trenutnog korisnika.
PATCH	/api/settings	Ažuriranje profila i postavki.
DELETE	/api/settings	Brisanje korisničkog računa (soft delete).



Metoda	Endpoint	Opis
GET	/api/startups	Pregled svih startupa.
POST	/api/startups	Kreiranje novog startupa.
GET	/api/user/startups	Startupi trenutnog korisnika.
GET	/api/startups/[id]/members	Lista članova tima startupa.
GET	/api/startups/[id]/invites	Lista aktivnih pozivnica.
GET	/api/team	Detalji vlastitog tima.
POST	/api/team/invite	Pozivanje novog člana u tim.
GET	/api/team/invite/[token]	Provjera valjanosti pozivnice.
POST	/api/team/invite/[token]	Prihvatanje pozivnice za tim.

💬 Komunikacija & Chat

```

sequenceDiagram
    participant Sender
    participant API
    participant DB
    participant Pusher
    participant Receiver

    Sender->>API: POST /messages/send

```

```

API->>DB: Store Message
API->>Pusher: Trigger 'new-message'
Pusher-->>Receiver: WebSocket Event
Receiver->>API: GET /messages/[id]
API-->>Receiver: Fetch History

```

Metoda	Endpoint	Opis
GET	/api/conversations	Lista svih konverzacija.
POST	/api/conversations	Kreiranje nove konverzacije.
GET	/api/conversations/[id]	Detalji pojedine konverzacije.
GET	/api/messages/[id]	Dohvat poruka konverzacije (paginirano).
POST	/api/messages/send	Slanje nove poruke.
POST	/api/contact	Slanje kontakt forme (support).
POST	/api/pusher/auth	Autentifikacija za WebSocket kanale.

🤖 AI & Dashboard

Metoda	Endpoint	Opis
POST	/api/chat	Interakcija s AI asistentom (OpenAI).
GET	/api/dashboard/stats	Glavne metrike za dashboard.
GET	/api/dashboard/activity	Lista nedavnih aktivnosti.
GET	/api/growth	Metrike rasta (chart data).
GET	/api/network	Pretraga mreže korisnika (Network).
GET	/api/locations	Proxy za pretragu lokacija (Nominatim).

💰 Plaćanja

```

sequenceDiagram
    participant User
    participant API
    participant DB

```

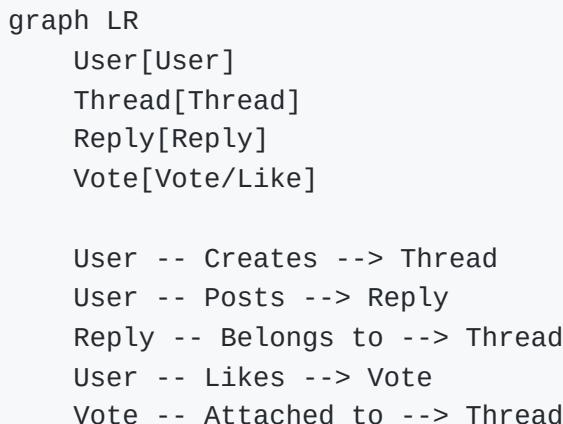
```

User->>API: POST /payments/calculate
API-->User: Fees (2.5%) & Net Amount
User->>API: POST /payments/send
API->>DB: Create Transaction (PENDING)
Note right of API: Webhook triggers status update
API-->User: Transaction Confirmed

```

Metoda	Endpoint	Opis
GET	/api/transactions	Povijest transakcija.
POST	/api/payments/calculate	Izračun naknada prije slanja.
POST	/api/payments/send	Izvršavanje plaćanja.

III Community (Threads & Roadmap)



Metoda	Endpoint	Opis
GET	/api/threads	Lista community diskusija.
POST	/api/threads	Kreiranje nove teme.
POST	/api/threads/[id]/like	Like/Unlike teme.
POST	/api/threads/[id]/replies	Odgovor na temu.
GET	/api/roadmap	Pregled feature requestova.
POST	/api/roadmap	Predlaganje nove funkcionalnosti.

POST	/api/roadmap/vote	Glasanje za funkcionalnost.
------	-------------------	-----------------------------

🛡 Admin & System

Metoda	Endpoint	Opis	Role
POST	/api/admin/reset-demo	Resetiranje demo podataka.	Admin
GET	/api/admin/users	Pregled svih korisnika.	Admin
PUT	/api/admin/users	Promjena uloge korisnika.	Admin
DELETE	/api/admin/users	Brisanje korisnika.	Admin
GET	/api/blog/posts	Upravljanje blog postovima.	Admin
GET	/api/creator/analytics/*	Creator analitika.	Admin
GET	/api/health	System health check.	Public
GET	/api/csrf-token	Dohvat CSRF tokena.	Auth
GET	/api/keys	Upravljanje API ključevima.	Auth

⚠ Rate Limiting

Sustav koristi rate limiting kako bi zaštitio API od zlouporabe.

Tier	Limit
Unauthenticated	10 zahtjeva / min
FREE	100 zahtjeva / min
PRO	1000 zahtjeva / min
GROWTH	10000 zahtjeva / min

⚠ Status Kodovi

Kod	Opis
200	OK - Uspješan zahtjev
201	Created - Resurs uspješno kreiran

400	Bad Request - Nevaljani podaci
401	Unauthorized - Nedostaje autentifikacija
402	Payment Required - Potrebno plaćanje (Antispam)
403	Forbidden - Nedovoljna prava pristupa
404	Not Found - Resurs nije pronađen
429	Too Many Requests - Prekoračen rate limit
500	Internal Server Error - Greška na serveru

Sigurnost

Sigurnosne Mjere

Mjera	Implementacija
Password Hashing	Argon2id ili bcryptjs (Cost factor 12) s automatskim saltanjem
Session Management	JWT tokeni s NextAuth (HS512 potpisivanje)
CSRF Protection	Middleware token validation
Rate Limiting	Redis-backed rate limiter
Input Sanitization	Zod validacija + HTML sanitization
SQL Injection	Prisma parameterized queries
XSS Prevention	React automatic escaping + CSP headers

Account Deletion Flow

```
sequenceDiagram
    participant U as User
    participant API as Server
    participant DB as Database
    participant CRON as Cleanup Job
```

```
U->>API: DELETE /api/settings (delete account)
API->>DB: SET deletedAt = NOW()
API->>DB: SET scheduledDeletionAt = NOW() + 30 days
API-->U: Account marked for deletion
```

Note over U, CRON: 30 dana grace period

```
U->>API: Login (within 30 days)
API->>DB: CLEAR deletedAt, scheduledDeletionAt
API-->U: Account reactivated!
```

Note over CRON: After 30 days

```
CRON->>DB: DELETE users WHERE scheduledDeletionAt < NOW()
CRON->>DB: CASCADE delete all related data
```

🐳 Konfiguracija i Deployment (Docker & CI/CD)

Projekt koristi kontejnerizaciju za konzistentnost i skalabilnost. Cjelokupna infrastruktura definirana je u `docker-compose.yml`, a aplikacija se gradi pomoću multi-stage `Dockerfile` skripte.

📦 Docker Compose Servisi

Naš stack se sastoji od 6 povezanih servisa unutar izolirane mreže `dfds-network`.

1. PostgreSQL Baza (`postgres`)

- **Image:** `postgres:15-alpine`
- **Uloga:** Primarno spremište podataka.
- **Konfiguracija:** Koristi `.env` varijable (`DB_USER`, `DB_PASSWORD`, `DB_NAME`) za sigurnost.
- **Volumeni:** Podaci su perzistentni putem `postgres_data` volumena.
- **Healthcheck:** Osigurava da je baza spremna (`pg_isready`) prije nego ovisni servisi startaju.

2. Redis Cache (`redis`)

- **Image:** `redis:7-alpine`
- **Uloga:** Cacheiranje sesija, rate limiting i Pub/Sub adapter za Soketi.

- **Konfiguracija:** Pokreće se s `appendonly yes` za trajnost podataka.
- **Volumeni:** `redis_data` za persistenciju.

3. Soketi WebSocket Server (`soketi`)

- **Image:** `quay.io/soketi/soketi:1.6-16-alpine`
- **Uloga:** Lokalna, brza zamjena za Pusher. Upravlja real-time komunikacijom (chat poruke, notifikacije).
- **Kompatibilnost:** Potpuno kompatibilan s Pusher protokolom (koristi `PUSHER_APP_KEY`, etc.).
- **Skalabilnost:** Dizajniran za visoke performanse i nisku latenciju.

4. Migrator (`migrator`)

- **Build Target:** `builder stage` iz Dockerfile-a.
- **Uloga:** Jednokratni servis koji se pokreće prije glavne aplikacije.
- **Zadatak:** Izvršava `prisma db push` i `seed.ts` kako bi sinkronizirao shemu baze i napunio je početnim podacima.
- **Profil:** `tools` (ne pokreće se automatski bez eksplisitnog poziva u `deploy.sh`).

5. Glavna Aplikacija (`app`)

- **Build Context:** Trenutni direktorij.
- **Uloga:** Next.js server koji servira frontend i API rute.
- **Port:** 3753 (interni i eksterni).
- **Ovisnosti:** Čeka (`depends_on`) da `postgres` i `redis` budu zdravi (`service_healthy`).
- **Resursi:** Ograničen na 1 CPU i 512MB RAM-a za predvidljivost.

6. Nginx Load Balancer (`nginx`)

- **Image:** `nginx:alpine`
- **Uloga:** Reverzni proxy i SSL terminacija (opcionalno za produkciju).
- **Profil:** `with-lb` (pokreće se samo uz `--profile with-lb`).
- **Portovi:** 80 (HTTP) i 443 (HTTPS).

Dockerfile Struktura (Multi-Stage Build)

Koristimo optimizirani multi-stage build proces kako bismo smanjili veličinu finalne slike i ubrzali deployment.

Stage	Opis
1. Deps	Instalira sistemske zavisnosti (<code>libc6-compat</code> , <code>openssl</code>) i <code>node_modules</code> za produkciju (<code>npm ci</code>). Cacheira se agresivno.
2. Builder	Kopira izvorni kod, generira Prisma klijent i pokreće <code>npm run build</code> . Ovdje se kreira optimizirani <code>.next/standalone</code> folder.
3. Runner	Finalna, lagana slika (<code>alpine</code>). Kopira samo nužne artefakte iz <code>Builder</code> faze (<code>standalone</code> , <code>static</code> , <code>public</code>). Pokreće se kao non-root korisnik (<code>nextjs</code>) radi sigurnosti.

Ključne Značajke:

- Sigurnost:** Aplikacija se vrti pod ne-privilegiranim korisnikom (`uid 1001`).
- Veličina:** Finalna slika je drastično manja jer ne sadrži `devDependencies` ni build alate.
- Standalone Mode:** Next.js Output Trace automatski detektira i pakira samo one datoteke koje su potrebne za runtime.

🔧 Deployment Skripta (@deploy.sh)

`deploy.sh` je centralizirani alat za upravljanje životnim ciklusom aplikacije. Automatizira instalaciju zavisnosti, konfiguraciju okoline, upravljanje kontejnerima i migracije baze podataka.

📋 Pregled Naredbi

Naredba	Opis	Primjer Korištenja
<code>install</code>	Instalira sistemske zavisnosti (Docker, Compose, Git, NVM).	<code>./deploy.sh install</code>
<code>setup</code>	Generira <code>.env</code> datoteku i sigurne ključeve (čuva postojeće).	<code>./deploy.sh setup</code>

<code>full</code>	Izvršava punu sekvencu: install → setup → start.	<code>./deploy.sh full</code>
<code>start</code>	Builda slike, pokreće migracije i starta produkcijski stack.	<code>./deploy.sh start</code>
<code>dev</code>	Pokreće razvojni mod s hot-reloadingom (koristi <code>docker-compose.dev.yml</code>).	<code>./deploy.sh dev</code>
<code>stop</code>	Sigurno zaustavlja sve containere i servise.	<code>./deploy.sh stop</code>
<code>clean</code>	Destruktivno: Briše sve containere, volume i podatke.	<code>./deploy.sh clean</code>
<code>status</code>	Prikazuje status aktivnih containera i portove.	<code>./deploy.sh status</code>
<code>logs</code>	Prikazuje live logove svih servisa.	<code>./deploy.sh logs</code>
<code>force-db-push</code>	Forsira ažuriranje baze uz mogući gubitak podataka (korisno za dev).	<code>./deploy.sh force-db-push</code>

⚙️ Automatizirani Procesi

Skripta interno rješava kompleksne zadaće kako bi deployment bio "jedan klik":

- 1. Detekcija Okoline:** Prepoznaje OS (Ubuntu, Debian, CentOS, Arch) i prilagođava instalacijske naredbe.
- 2. Upravljanje Tajnama:**
 - Automatski generira jake lozinke za bazu i `NEXTAUTH_SECRET`.
 - Pri ponovnom pokretanju `setup`, **čuva postojeće ključeve** (OpenAI, Resend, DB credentials) kako ne biste izgubili konfiguraciju.
- 3. Migracije Baze:**
 - Čeka da PostgreSQL postane dostupan.
 - Pokreće `prisma db push` i `seed` skripte u izoliranom okruženju prije pokretanja aplikacije.
- 4. Load Balancing (Opcionalno):**
 - Podržava `start-lb` za pokretanje Nginx load balanca ispred aplikacije (korisno za skaliranje).

🚀 Scenariji Korištenja

Prvi Deployment (Fresh Install)

```
# Kloniraj repo i pokreni full setup
git clone ...
cd dfds
./deploy.sh full
```

Lokalni Razvoj (Development)

```
# Pokreće app s mountanim volume-ima za live editiranje koda
./deploy.sh dev
```

Ažuriranje Producije

```
git pull
./deploy.sh restart
# Ili za puni rebuild:
./deploy.sh start
```

Detaljni Popis Datoteka

Ovo je referenca svih ključnih datoteka u projektu, grupiranih po direktorijima.

Aplikacija (/app)

Javne Stranice

Datoteka	Opis
page.tsx	Glavna naslovna stranica (Landing Page). Sadrži Hero sekciju, vizualizaciju mreže i footer.
about/page.tsx	"O nama" stranica. Prikazuje misiju tima Cloudzz i viziju projekta.
contact/page.tsx	Kontakt stranica s linkovima na društvene mreže i email.

<code>careers/page.tsx</code>	Stranica za karijere (trenutno statična, poziva na slanje životopisa).
<code>privacy/page.tsx</code>	Izjava o privatnosti i rukovanju podacima.
<code>terms/page.tsx</code>	Uvjeti korištenja platforme.
<code>docs/page.tsx</code>	Renderira ovu tehničku dokumentaciju (<code>DFDS_Dokumentacija.md</code>) unutar same aplikacije.
<code>layout.tsx</code>	Glavni layout aplikacije. Uključuje providere (<code>AuthProvider</code> , <code>PostHogProvider</code> , <code>ThemeProvider</code>) i globalne stilove.
<code>globals.css</code>	Globalni Tailwind CSS stilovi i custom animacije.
<code>not-found.tsx</code>	Custom 404 stranica.
<code>error.tsx</code>	Globalni error boundary za hvatanje neočekivanih grešaka.

Autentifikacija

Datoteka	Opis
<code>login/page.tsx</code>	Stranica za prijavu. Podržava email/password i "Demo Account" prečac.
<code>join/page.tsx</code>	Multi-step registracija. Korisnik bira ulogu (Developer, Founder, Investor) i unosi detalje profila.
<code>verify-email/page.tsx</code>	Stranica za unos 6-znamenkastog verifikacijskog koda.
<code>verify-email/VerifyEmailContent.tsx</code>	Klijentska logika za verifikaciju emaila (forma, api poziv).
<code>reset-password/page.tsx</code>	Forma za postavljanje nove lozinke (zahtijeva token iz emaila).

Dashboard (Zaštićeno)

Datoteka	Opis
<code>dashboard/layout.tsx</code>	Layout za dashboard. Sadrži bočnu traku, gornju navigaciju i provjeru sesije.

dashboard/page.tsx	Glavni pregled dashboarda. Prikazuje widgete ovisno o ulozi korisnika.
dashboard/analytics/page.tsx	Stranica za naprednu analitiku (u izradi).
dashboard/api-access/page.tsx	Upravljanje API ključevima za developere.
dashboard/calculator/page.tsx	Kalkulator equity dilucije.
dashboard/developer/page.tsx	Specifični dashboard za developere (zarada, projekti).
dashboard/investor/page.tsx	Specifični dashboard za investitore (portfolio, deal flow).
dashboard/startup/page.tsx	Specifični dashboard za osnivače (metrike, tim).
dashboard/investors/page.tsx	Pretraga i grid prikaz investitora.
dashboard/members/page.tsx	Upravljanje članovima tima i pozivnicama.
dashboard/messages/page.tsx	Glavno sučelje za chat (inbox + konverzacija).
dashboard/network/page.tsx	Mreža developera s pretragom i filtrima.
dashboard/payments/page.tsx	Sučelje za slanje novca i povijest transakcija.
dashboard/roadmap/page.tsx	Prikaz roadmapa s mogućnošću glasanja za značajke.
dashboard/settings/page.tsx	Postavke profila, notifikacija i sigurnosti (uključujući brisanje računa).
dashboard/startups/page.tsx	Preglednik svih startupa na platformi.
dashboard/threads/page.tsx	Community forum za diskusije.

Ostalo

Datoteka	Opis
api-docs/page.tsx	Prikaz API dokumentacije.
blog/page.tsx	Lista blog članaka.
blog/[slug]/page.tsx	Pojedinačni blog članak.

blog/admin/page.tsx	Admin sučelje za upravljanje blogom.
creator/dashboard/page.tsx	Admin dashboard s KPI metrikama sustava.
discover/page.tsx	Javni pretraživač startupa i talenata (bez prijave).
paywall-demo/page.tsx	Demo stranica za testiranje crypto plaćanja.
profile/[id]/page.tsx	Javni profil korisnika.

⚡ API Rute (/app/api)

Auth & Korisnici

Datoteka	Opis
auth/[^nextauth]/route.ts	NextAuth.js konfiguracija i handleri.
auth/reset-password/request/route.ts	Slanje emaila za reset lozinke.
auth/reset-password/confirm/route.ts	Promjena lozinke pomoću tokena.
register/route.ts	Registracija novog korisnika i kreiranje profila.
verify-email/route.ts	Verifikacija email adrese.
resend-verification/route.ts	Ponovno slanje verifikacijskog emaila.
profiles/[id]/route.ts	Dohvat javnog profila.
settings/route.ts	Dohvat i ažuriranje postavki korisnika.

Startipi & Timovi

Datoteka	Opis
startups/route.ts	CRUD za startupe.
startups/[id]/members/route.ts	Dohvat članova tima startupa.
startups/[id]/invites/route.ts	Dohvat aktivnih pozivnica za startup.
team/route.ts	Dohvat vlastitog tima.

team/invite/route.ts	Slanje pozivnice za tim.
team/invite/[token]/route.ts	Validacija i prihvatanje pozivnice.
applications/join/route.ts	Slanje prijave za pridruživanje.
user/startups/route.ts	Dohvat startupa trenutnog korisnika.

Komunikacija

Datoteka	Opis
messages/send/route.ts	Slanje poruke (s provjerom plaćanja).
messages/conversations/route.ts	Lista konverzacije.
messages/[conversationId]/route.ts	Dohvat poruka konverzacije.
conversations/route.ts	Upravljanje konverzacijama.
contact/route.ts	Slanje kontakt forme.
pusher/auth/route.ts	Autentifikacija za privatne WebSocket kanale.
threads/route.ts	Upravljanje temama foruma.
threads/[id]/replies/route.ts	Odgovaranje na teme.
threads/[id]/like/route.ts	Lajkanje tema.

Sustav & Admin

Datoteka	Opis
admin/users/route.ts	Admin upravljanje korisnicima.
admin/reset-demo/route.ts	Resetiranje demo podataka (samo dev/admin).
keys/route.ts	Upravljanje API ključevima.
health/route.ts	Health check endpoint.
cookie-consent/route.ts	Spremanje privole za kolačiće.
documentation/route.ts	Servira JSON specifikaciju API-ja.

locations/route.ts

Proxy za Nominatim (OpenStreetMap) pretragu.

Ostalo

Datoteka	Opis
chat/route.ts	OpenAI chat endpoint.
payments/calculate/route.ts	Izračun naknada za transakcije.
payments/send/route.ts	Izvršavanje transakcije.
transactions/route.ts	Povijest transakcija.
dashboard/stats/route.ts	Agregirane statistike za dashboard.
dashboard/activity/route.ts	Nedavne aktivnosti korisnika.
growth/route.ts	Metrike rasta za grafove.
network/route.ts	Pretraga mreže korisnika.
roadmap/route.ts	Upravljanje roadmapom.
roadmap/vote/route.ts	Glasanje na roadmapu.
blog/posts/route.ts	CRUD za blog postove.
v1/**	Javni API endpointi (startups, ingest, funding, portfolio).

ksam Komponente (/components)

UI Primitivi (/ui)

Datoteka	Opis
GlassCard.tsx	Kartica s efektom stakla (blur, border).
Button.tsx	Gumb s raznim varijantama (default, ghost, glass).
Input.tsx , Textarea.tsx	Form inputi stilizirani za dark mode.
Badge.tsx	Oznaka za status (boje po ulozi/statusu).

Avatar.tsx	Prikaz korisničkog avatara ili inicijala.
Dialog.tsx	Modalni dijaloški okvir.
Select.tsx	Dropdown izbornik.
Switch.tsx , Checkbox.tsx	Toggle i checkbox kontrole.
Section.tsx	Wrapper za sekcije stranice.
StatCard.tsx , ChartCard.tsx	Komponente za prikaz metrika i grafova.
BentoGrid.tsx	Grid layout za prikaz sadržaja.
BackgroundGrid.tsx , EnhancedBackground.tsx	Pozadinski efekti.
Beams.tsx	Three.js animacija svjetlosnih zraka.
DynamicIsland.tsx	Plutajuća navigacija.
InfiniteMovingCards.tsx	Animirani scroll kartica (testimonials).
LocationAutocomplete.tsx	Input za pretragu lokacija.
SkillsAutocomplete.tsx	Input za unos vještina s prijedlozima.
MetricsChart.tsx	Recharts wrapper za grafove.
MicroExpander.tsx	Gumb koji se širi na hover.
ThemeToggle.tsx	Prekidač za tamnu/svjetlu temu.
CommandMenu.tsx	CMD+K izbornik za brzu navigaciju.
CookieConsent.tsx	Banner za privolu kolačića.

Funkcionalne Komponente

Kategorija	Datoteke	Opis
AI	AiAssistant.tsx	Chat widget s AI asistentom.

Auth	<code>AuthButton.tsx</code> , <code>ReferralPaywall.tsx</code>	Gumbi za prijavu, paywall za zaključani sadržaj.
Dashboard	<code>DashboardSearch.tsx</code> , <code>FounderDashboard.tsx</code> , <code>GrowthDashboard.tsx</code> , <code>InvestorGrid.tsx</code> , <code>NetworkGrid.tsx</code> , <code>NetworkSearch.tsx</code> , <code>UserNav.tsx</code>	Widgeti specifični za dashboard.
Features	<code>EquitySimulator.tsx</code> , <code>SimulatorControls.tsx</code> , <code>RoadmapView.tsx</code>	Kalkulator equity-ja i prikaz roadmapa.
Landing	<code>CallToAction.tsx</code> , <code>FeatureCards.tsx</code> , <code>HeroContent.tsx</code> , <code>HeroSection.tsx</code> , <code>HowItWorks.tsx</code> , <code>ModernFooter.tsx</code> , <code>NetworkPulse.tsx</code>	Sekcije naslovne stranice.
Messaging	<code>ConversationView.tsx</code> , <code>MessageInbox.tsx</code> , <code>PaymentRequiredModal.tsx</code>	Komponente za chat sustav.
Payments	<code>PaymentModal.tsx</code> , <code>TransactionHistory.tsx</code> , <code>UserSearch.tsx</code>	Modal za plaćanje i povijest.
Startups	<code>ConnectButton.tsx</code> , <code>CreateStartupButton.tsx</code> , <code>StartupsSearch.tsx</code>	Akcije vezane uz startupe.
Providers	<code>AuthProvider.tsx</code> , <code>PostHogProvider.tsx</code> , <code>ThemeProvider.tsx</code>	Context provideri.
Demo	<code>DemoControls.tsx</code> , <code>Paywall.tsx</code>	Kontrole za demo mod.

Biblioteke i Alati (`/lib`)

Datoteka	Opis
<code>api-key-auth.ts</code>	Middleware za validaciju API ključeva.
<code>api-key-utils.ts</code>	Generiranje i hashing API ključeva.

<code>api-proxy.ts</code>	Proxy za sigurno pozivanje internih API-ja.
<code>api-security.ts</code>	Helperi za rate limiting i provjeru sesije u API rutama.
<code>auth.ts</code>	Konfiguracija NextAuth.js providera i callbacka.
<code>conversation-utils.ts</code>	Pomoćne funkcije za upravljanje chatom.
<code>crypto-utils.ts</code>	Maskiranje crypto adresa.
<code>csrf.ts</code>	Generiranje i validacija CSRF tokena.
<code>dashboard-queries.ts</code>	Optimizirani upiti za dashboard statistiku.
<code>demo-utils.ts</code>	Logika za resetiranje demo podataka.
<code>email.ts</code>	Slanje emailova putem Resenda.
<code>env.ts</code>	Validacija environment varijabli (Zod).
<code>error-handler.ts</code>	Globalni error handler za API.
<code>logger.ts</code>	Strukturirano logiranje s redakcijom osjetljivih podataka.
<code>matchmaker.ts</code>	Algoritam za matching (mock vector search).
<code>math/dilution.ts</code>	Matematičke funkcije za equity kalkulator.
<code>payment-utils.ts</code>	Izračuni naknada i formatiranje valuta.
<code>posthog-server.ts</code>	PostHog klijent za serversku stranu.
<code>prisma.ts</code>	Prisma klijent instanca.
<code>pusher.ts</code>	Pusher/Soketi konfiguracija (klijent i server).
<code>rate-limit.ts</code>	Redis rate limiter konfiguracija.
<code>sanitize.ts</code>	Sanitizacija HTML-a i teksta.
<code>user-cleanup.ts</code>	Logika za brisanje isteklih (soft-deleted) korisnika.
<code>utils.ts</code>	Opći utility helperi (cn).
<code>validations.ts</code>	Zod sheme za validaciju unosa.

Hookovi (`/hooks`)

Datoteka	Opis
<code>use-debounce.ts</code>	Hook za odgodu izvršavanja (npr. pretraga).
<code>useConversationSubscription.ts</code>	Hook za real-time chat (Pusher subscription).

Skripte (`/scripts`)

Datoteka	Opis
<code>CreateThread.ts</code>	Kreira testni thread.
<code>GenerateApiKey.ts</code>	Generira API ključ za testiranje.
<code>GetVerificationCode.ts</code>	Dohvaća verifikacijski kod iz baze.
<code>RepairTeams.ts</code>	Popravlja startupe koji nemaju tim.
<code>UpdateUserRole.ts</code>	Ažurira ulogu korisnika.
<code>VerifyUser.ts</code>	Ručno verificira email korisnika.
<code>smoke-test.sh</code>	Bash skripta za brzu provjeru zdravlja sustava.
<code>verify-ingestion-db.ts</code>	Testira direktni upis metrika u bazu.
<code>verify-message-fetching.ts</code>	Testira dohvat poruka iz baze.
<code>verify-middleware.ts</code>	Testira middleware pravila.
<code>verify-production-upgrade.ts</code>	Provjerava integritet podataka.
<code>verify-user-deletion.ts</code>	Testira kaskadno brisanje korisnika.

Testovi (`/tests`)

E2E (`/tests/e2e`)

Datoteka	Opis
<code>components/ai.test.tsx</code>	Testira AI chat komponentu.
<code>components/auth.test.tsx</code>	Testira auth gumbu.

components/dashboard.test.tsx	Testira dashboard widgete.
components/landing.test.tsx	Testira landing page sekcije.
components/messaging.test.tsx	Testira chat sučelje.
components/payments.test.tsx	Testira payment modal.
components/transaction-history.test.tsx	Testira povijest transakcija.
components/ui.test.tsx	Testira osnovne UI komponente.
components/verify-email-content.test.tsx	Testira formu verifikacije.
dashboard/pages.test.tsx	Testira stranice dashboarda.

Integration (`/tests/integration`)

Datoteka	Opis
api-admin-users.test.ts	Testira admin API za korisnike.
api-analytics-internal.test.ts	Testira interni analytics API.
api-blog-posts-id.test.ts	Testira operacije nad pojedinim blog postom.
api-blog-posts.test.ts	Testira listanje i kreiranje blog postova.
api-chat.test.ts	Testira chat API endpoint.
api-documentation.test.ts	Testira docs endpoint.
api-gap-fill.test.ts	Testira razne rubne slučajeve API-ja.
api-health.test.ts	Testira health check.
api-ingest-security.test.ts	Testira sigurnost ingest API-ja.
api-ingest.test.ts	Testira logiku ingestije podataka.
api-key-auth.test.ts	Testira auth middleware za API ključeve.
api-key-timing-attack.test.ts	Testira otpornost na timing napade.
api-keys.test.ts	Testira upravljanje ključevima.

<code>api-messaging.test.ts</code>	Testira messaging API.
<code>api-phase1.test.ts</code>	Testira inicijalne API rute.
<code>api-phase2.test.ts</code>	Testira dashboard API rute.
<code>api-phase3.test.ts</code>	Testira startup API rute.
<code>api-phase4.test.ts</code>	Testira komunikacijske API rute.
<code>api-phase5.test.ts</code>	Testira content i payment API rute.
<code>api-phase6.test.ts</code>	Testira growth API rute.
<code>api-phase7.test.ts</code>	Testira admin i v1 API rute.
<code>api-team-invite.test.ts</code>	Testira invite flow API.
<code>auth-verification.test.ts</code>	Testira logiku verifikacije.
<code>lib-api-key-auth.test.ts</code>	Testira lib funkcije za API auth.
<code>payment-idempotency.test.ts</code>	Testira idempotenciju plaćanja.
<code>security.test.ts</code>	Generalni sigurnosni testovi.

Unit (`/tests/unit`)

Datoteka	Opis
<code>lib-api-key.test.ts</code>	Unit testovi za API key utilse.
<code>lib-email.test.ts</code>	Unit testovi za slanje emaila.
<code>lib-env.test.ts</code>	Unit testovi za env validaciju.
<code>lib-error-handler.test.ts</code>	Unit testovi za error handling.
<code>lib-logger.test.ts</code>	Unit testovi za logger.
<code>lib-payment.test.ts</code>	Unit testovi za payment utilse.
<code>lib-rate-limit.test.ts</code>	Unit testovi za rate limiter.
<code>lib-security.test.ts</code>	Unit testovi za sigurnosne utilse.
<code>lib-utils.test.ts</code>	Unit testovi za opće utilse.

math/dilution.test.ts	Unit testovi za matematiku dilucije.
setup.ts	Setup datoteka za testove.
xss-sanitization.test.ts	Unit testovi za XSS sanitizaciju.

Kako Pokrenuti

Preduvjeti

- Node.js 18+
- PostgreSQL 14+
- Redis (opcionalno, za full real-time)
- npm (preporučeno zbog package-lock.json)

Koraci

```
# 1. Klonirajte repozitorij
git clone https://github.com/team-cloudzz/dfds.git
cd dfds

# 2. Instalirajte dependencies
npm ci # ili npm install

# 3. Konfigurirajte environment varijable
cp env.example .env
# Uredite .env s vašim podacima

# 4. Inicijalizirajte bazu
npx prisma db push
npx prisma db seed

# 5. Pokrenite development server
npm run dev

# 6. Otvorite u pregledniku
open http://localhost:3000
```

Environment Varijable

```
# Database
DATABASE_URL="postgresql://user:pass@localhost:5432/dfds"

# NextAuth
NEXTAUTH_SECRET="your-secret-key"
NEXTAUTH_URL="http://localhost:3000"

# OAuth Providers
GITHUB_ID="..."
GITHUB_SECRET="..."
GOOGLE_CLIENT_ID="..."
GOOGLE_CLIENT_SECRET="..."

# Services
PUSHER_APP_ID="..."
PUSHER_KEY="..."
PUSHER_SECRET="..."
RESEND_API_KEY="..."
OPENAI_API_KEY="..."
POSTHOG_KEY="..."

# Redis (optional)
REDIS_URL="redis://localhost:6379"
```

Docker Deployment

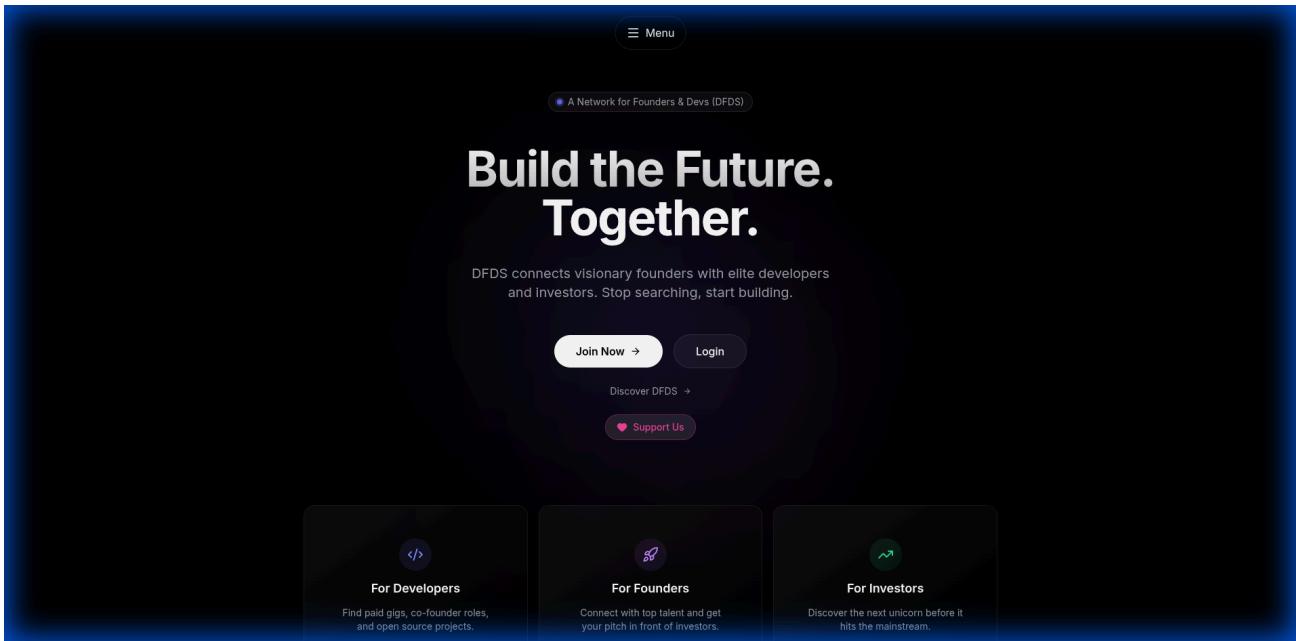
```
# Build i pokreni sve servise
docker-compose up -d

# Ili koristi deploy skriptu
./deploy.sh full
```



Naslovna Stranica (Landing Page)

Prva stranica koju posjetitelji vide. Dizajnirana za konverziju.



Sekcije:

- 1. Hero Section** - Glavni naslov, opis i CTA gumbi
- 2. How It Works** - 3-step objašnjenje platforme
- 3. Call To Action** - Registracijski CTA
- 4. Modern Footer** - Linkovi, social media, copyright

📞 Kontakt i Podrška

- Email: team@cloudzz.dev

Hvala što koristite DFDS!

— Tim Cloudzz

Verzija dokumentacije: 2.0.0

Zadnje ažuriranje: Siječanj 2026

🧩 Komponente Sučelja (@components)

Frontend arhitektura temelji se na modularnim i ponovno iskoristivim React komponentama. Organizirane su prema funkcionalnosti unutar `@components` direktorija.

Hijerarhija Komponenti (Primjer Dashboarda)

```
graph TD
    Page[Dashboard Page]
    Layout[Dashboard Layout]

    Layout --> Sidebar[Sidebar Nav]
    Layout --> Header[Header]
    Layout --> AiAssistant[AI Assistant Widget]

    Header --> Search[Dashboard Search]
    Header --> UserNav[User Menu]

    Page --> StatsGrid[Stats Grid]
    Page --> Charts[Growth Charts]

    StatsGrid --> StatCard1[StatCard]
    StatsGrid --> StatCard2[StatCard]

    Charts --> ChartCard1[ChartCard]
```

AI Komponente (/ai)

Komponenta	Opis
AiAssistant	Glavni chat widget koji pluta na ekranu. Koristi <code>react-markdown</code> za renderiranje odgovora i podržava "Quick Actions" (npr. analiza pitcha). Povezan s OpenAI API-jem.

Autentifikacija (/auth)

Komponenta	Opis
AuthButton	Gumb za prijavu/odjavu s vizualnim indikatorom stanja.
ReferralPaywall	"Soft paywall" komponenta koja zaključava sadržaj dok korisnik ne pozove prijatelje ili ne nadogradi plan.

Dashboard Elementi (/dashboard)

Komponenta	Opis
------------	------

StatCard	Prikaz pojedinačne metrike (npr. Revenue) s malim trend indikatorom i sparkline grafom.
ChartCard	Wrapper oko <code>recharts</code> biblioteke za prikaz linijskih, stupčastih i površinskih grafova.
GrowthDashboard	Složena komponenta koja prikazuje analitiku rasta (prihodi, korisnici) s mogućnošću filtriranja po vremenu.
FounderDashboard	Specijalizirani dashboard za osnivače s metrikom "burn rate" i "runway".
UserNav	Dropdown izbornik za korisnički profil, postavke i odjavu.
NetworkGrid & InvestorGrid	Grid prikaz korisnika s karticama, avatarima i akcijskim gumbima.

🚀 Landing Page (/landing)

Komponenta	Opis
HeroSection	Glavni naslovni dio s animiranim tekstom i CTA gumbima.
NetworkPulse	Vizualizacija aktivnosti u stvarnom vremenu ("Live Activity Feed").
FeatureCards	Prikaz ključnih mogućnosti platforme u gridu.
ModernFooter	Responzivno podnožje s linkovima i formom za newsletter.

💬 Messaging (/messaging)

Komponenta	Opis
ConversationView	Glavni prozor chata. Sadrži listu poruka, input polje i header s informacijama o sugovorniku. Podržava real-time ažuriranja.
MessageInbox	Lista svih konverzacija s prikazom zadnje poruke i statusom pročitanosti.

PaymentRequiredModal	Modal koji se pojavljuje kada je potrebno platiti za slanje poruke (anti-spam mjera).
----------------------	---

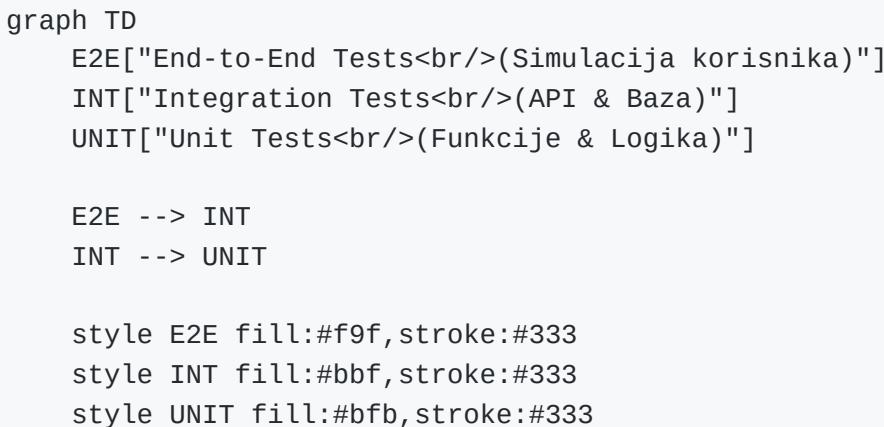
💳 Plaćanja (/payments)

Komponenta	Opis
PaymentModal	Dijalog za slanje novca. Računa provizije i odabire metodu plaćanja (Kripto, Kartica).
TransactionHistory	Tablica svih transakcija s filtriranjem (poslano/primljeno).
UserSearch	Input polje s autocomplete funkcionalnošću za pronalaženje primatelja plaćanja.

🧪 Testiranje i CI/CD Pipeline (@tests)

Kvaliteta koda osigurana je kroz sveobuhvatan sustav testiranja koji pokriva sve razine aplikacije, od pojedinačnih funkcija do korisničkih flowova.

⚠️ Strategija Testiranja (Testing Pyramid)



1. Unit Testovi (@tests/unit/**)

Testiraju izolirane dijelove logike bez vanjskih ovisnosti.

Datoteka	Opis Testa
----------	------------

<code>lib-api-key.test.ts</code>	Provjerava generiranje, hashing i validaciju formata API ključeva.
<code>lib-email.test.ts</code>	Testira generiranje verifikacijskih kodova i mockira slanje emailova putem Resenda.
<code>lib-env.test.ts</code>	Osigurava da aplikacija ne starta bez nužnih varijabli (<code>DATABASE_URL</code> , <code>NEXTAUTH_SECRET</code>).
<code>lib-error-handler.test.ts</code>	Verificira da se greške (npr. ZodError, PrismaError) ispravno transformiraju u HTTP odgovore.
<code>lib-logger.test.ts</code>	Provjerava formatiranje logova i redakciju osjetljivih podataka (npr. password).
<code>lib-payment.test.ts</code>	Testira matematiku izračuna provizija (2.5%) i formatiranje valuta.
<code>lib-rate-limit.test.ts</code>	Provjerava ekstrakciju IP adresa (<code>x-forwarded-for</code>) i logiku limitiranja zahtjeva.
<code>lib-security.test.ts</code>	Testira pomoćne funkcije <code>ensureAuth</code> i <code>ensureRateLimit</code> .
<code>lib-utils.test.ts</code>	Testira utility funkcije poput <code>cn</code> (Tailwind merge) i sanitizacije teksta.
<code>math/dilution.test.ts</code>	Verificira točnost izračuna equity dilucije i post-money valuacije.
<code>setup.ts</code>	Konfigurira globalne mockove (NextAuth, <code>localStorage</code>) za testno okruženje.
<code>xss-sanitization.test.ts</code>	Provjerava da <code>sanitizeHtml</code> uklanja maliciozne skripte i tagove.

2. Integracijski Testovi (@tests/integration/**)

Testiraju interakciju između API-ja, baze podataka i servisa.

Datoteka	Opis Testa
<code>api-admin-users.test.ts</code>	Provjerava da samo admini mogu listati, mijenjati i brisati korisnike.

<code>api-analytics-internal.test.ts</code>	Testira pristup internim analitičkim podacima (samo admin).
<code>api-blog-posts-id.test.ts</code>	Testira dohvat, ažuriranje i brisanje pojedinačnih blog postova.
<code>api-blog-posts.test.ts</code>	Testira kreiranje i listanje blog postova.
<code>api-chat.test.ts</code>	Verificira formatiranje poruka za OpenAI i odgovor API-ja.
<code>api-documentation.test.ts</code>	Provjerava da <code>/api/documentation</code> vraća ispravnu specifikaciju.
<code>api-gap-fill.test.ts</code>	Pokriva rubne slučajeve u analitici i pozivnicama za startup.
<code>api-health.test.ts</code>	Provjerava da <code>/api/health</code> vraća status 200.
<code>api-ingest-security.test.ts</code>	Testira sigurnost ingestije metrika (provjera API ključa, rate limit).
<code>api-ingest.test.ts</code>	Testira sinkronizaciju timova i funding podataka putem eksternog API-ja.
<code>api-key-auth.test.ts</code>	Detaljno testira middleware za autentifikaciju API ključeva.
<code>api-key-timing-attack.test.ts</code>	Sigurnost: Potvrđuje da provjera ključeva ne "curi" informacije kroz vrijeme izvršavanja.
<code>api-keys.test.ts</code>	Testira CRUD operacije za upravljanje API ključevima korisnika.
<code>api-messaging.test.ts</code>	Provjerava dohvat konverzacija i poruka, te kontrolu pristupa (samo sudionici).
<code>api-phase1.test.ts</code>	Testira osnovne flowove: zdravlje sustava, verifikacija emaila.
<code>api-phase2.test.ts</code>	Testira dashboard API-je: statistika, aktivnost, postavke.
<code>api-phase3.test.ts</code>	Testira upravljanje startupima i timovima (invite flow).

<code>api-phase4.test.ts</code>	Testira komunikacijske značajke (chat, threadovi).
<code>api-phase5.test.ts</code>	Testira blog i finansijske transakcije.
<code>api-phase6.test.ts</code>	Testira growth značajke (kontakt forma, likeovi, reply).
<code>api-phase7.test.ts</code>	Testira admin funkcije i javne v1 endpointe.
<code>api-team-invite.test.ts</code>	Fokusirani test za logiku prihvatanja pozivnica u tim.
<code>auth-verification.test.ts</code>	Provjerava da ne-verificirani korisnici ne mogu pristupiti zaštićenim rutama.
<code>payment-idempotency.test.ts</code>	Sigurnost: Osigurava da se ista transakcija ne naplati dvaput.
<code>security.test.ts</code>	Generalni sigurnosni testovi (npr. zabrana starih demo lozinki).

3. End-to-End (E2E) Testovi (@tests/e2e/**)

Simuliraju stvarna korisnička ponašanja renderiranjem komponenti.

Datoteka	Opis Testa
<code>components/ai.test.tsx</code>	Testira interakciju s AI chatom (unos pitanja, prikaz odgovora).
<code>components/auth.test.tsx</code>	Provjerava prikaz gumba za prijavu/odjavu.
<code>components/dashboard.test.tsx</code>	Testira renderiranje dashboard widgeta i navigacije.
<code>components/landing.test.tsx</code>	Provjerava ključne elemente landing stranice (CTA, Hero, Footer).
<code>components/messaging.test.tsx</code>	Testira inbox i prozor za razgovor (slanje poruke).
<code>components/payments.test.tsx</code>	Testira modal za plaćanje i izračun naknada.

<code>components/transaction-history.test.tsx</code>	Provjerava prikaz povijesti transakcija.
<code>components/ui.test.tsx</code>	Testira ispravno renderiranje osnovnih UI komponenti (Avatar, Card, Chart).
<code>components/verify-email-content.test.tsx</code>	Testira formu za unos verifikacijskog koda.
<code>dashboard/pages.test.tsx</code>	Integracijski test cijelih stranica dashboarda (Settings, Investor view).

🚀 CI/CD Pipeline

Svaki `git push` pokreće automatizirani pipeline koji osigurava da novi kod ne narušava postojeću funkcionalnost.

```

sequenceDiagram
    participant Dev as Developer
    participant Git as GitHub Repo
    participant CI as CI Runner

    Dev->>Git: Push Code
    Git->>CI: Trigger Workflow

    rect rgb(240, 248, 255)
        Note right of CI: 1. Build & Lint
        CI->>CI: npm install
        CI->>CI: npm run lint (ESLint)
        CI->>CI: npm run build (Next.js)
    end

    rect rgb(255, 240, 245)
        Note right of CI: 2. Test Suite
        CI->>CI: npm run test:unit
        CI->>CI: npm run test:integration
        CI->>CI: npm run test:e2e
    end

    alt Success
        CI-->>Git: ✅ All Checks Passed
        Git-->>Dev: Ready to Merge
    else Failure
        CI-->>Git: ❌ Checks Failed
    end

```

```
Git-->>Dev: Fix Errors  
end
```

Alati za Testiranje

Alat	Svrha
Vitest	Glavni test runner. Brz, kompatibilan s Jest-om, podržava TypeScript.
React Testing Library	Renderiranje React komponenti u E2E testovima.
MSW (Mock Service Worker)	Mockiranje mrežnih zahtjeva (u planu za buduća proširenja).
GitHub Actions	Orkestracija CI/CD pipeline-a.