**What is Supabase?**

**Supabase is an open-source alternative to Firebase** that provides backend-as-a-service using PostgreSQL instead of NoSQL. It's essentially a suite of tools that helps developers build applications faster by handling the backend infrastructure.

**Core Components**

**1. PostgreSQL Database**

* **Fully relational database** (unlike Firebase's NoSQL)
* **SQL support** - you can write complex queries
* **Row-level security** for data protection
* **Extensions** for additional functionality

**2. Auto-generated APIs**

javascript

// REST API example

// GET all users

fetch('https://your-project.supabase.co/rest/v1/users', {

headers: {

'apikey': 'your-anon-key'

}

})

// Realtime example

const subscription = supabase

.from('users')

.on('UPDATE', payload => {

console.log('User updated!', payload)

})

.subscribe()

**3. Authentication**

* Email/password auth
* Social logins (Google, GitHub, etc.)
* Magic links
* Phone authentication
* JWT tokens

**4. Storage**

* File storage and CDN
* Image transformations
* Access control

**5. Realtime Subscriptions**

* Listen to database changes in real-time
* Perfect for chat apps, live dashboards

**Key Features**

**🔥 Instant Setup**

bash

# Start locally in minutes

docker-compose up -d

# Or use cloud version

**🔄 Real-time by Default**

javascript

// Listen to database changes

supabase

.from('todos')

.on('INSERT', payload => {

console.log('New todo!', payload.new)

})

.subscribe()

**🔐 Built-in Security**

sql

-- Row Level Security policies

CREATE POLICY "Users can view own profile"

ON profiles FOR SELECT

USING (auth.uid() = id);

**📱 Multiple Client Libraries**

javascript

// JavaScript

const { data, error } = await supabase

.from('countries')

.select('name')

// Python

data = supabase.table('countries').select('name').execute()

// Flutter

final data = await supabase

.from('countries')

.select('name');

**Supabase vs Firebase**

| Aspect | Supabase | Firebase |
| --- | --- | --- |
| **Database** | PostgreSQL (SQL) | Firestore (NoSQL) |
| **API** | REST + GraphQL | Proprietary SDKs |
| **Realtime** | PostgreSQL replication | Proprietary |
| **Pricing** | Postgres-based | Usage-based |
| **Open Source** | ✅ Yes | ❌ No |

**Common Use Cases**

**1. Web Applications**

javascript

// Full-stack app example

const { data: user } = await supabase.auth.getUser()

const { data: posts } = await supabase

.from('posts')

.select(`

\*,

profiles (username, avatar\_url)

`)

.eq('user\_id', user.id)

**2. Mobile Apps**

dart

// Flutter example

await supabase

.from('todos')

.insert({'task': 'Buy milk', 'user\_id': userId});

**3. Real-time Dashboards**

javascript

// Live analytics

supabase

.from('analytics')

.on('INSERT', payload => {

updateChart(payload.new)

})

.subscribe()

**Architecture Overview**

text

Your Frontend App

↓

Supabase Client Library

↓

Supabase API Layer

↓

PostgreSQL Database ← Realtime ← Auth ← Storage

**Why Developers Love Supabase**

1. **Uses PostgreSQL** - familiar, powerful, SQL-standard
2. **Open Source** - self-hostable, transparent
3. **Real-time** - built-in, no extra setup
4. **Great DX** - excellent documentation and tools
5. **Generous Free Tier** - perfect for startups and side projects

**Getting Started Flow**

1. **Create Project** (cloud or local)
2. **Create Tables** in the SQL editor
3. **Enable Auth** if needed
4. **Connect Client** to your frontend
5. **Build Features** using the APIs

**In essence: Supabase takes the powerful PostgreSQL database and wraps it with easy-to-use APIs, authentication, and real-time capabilities, making backend development much faster and more accessible.**

**Question: What's the Difference Between Supabase and Firebase?  
  
Question: Bro What Supabase Storage Bucket can do? Just like i created only storage bucket here "profile-pictures Navigate Reload View Upload files Create folder Select all 1 files profile\_6dc80ed4-9be0-4195-8cce-1beab1284463.jpg"  
  
Question: Bro What Supabase Edge Function can do? Just like i created only Supabase Edge Function here "Edge Functions Deploy edge functions to handle complex business logic Docs Examples Deploy a new function Name URL Last updated Deployments create-profile-on-signup https://zzmlainpucbkvcrtyrrg.supabase.co/functions/v1/create-profile-on-signup a day ago 1"  
  
Question:** **Bro What Auth Hooks can do? Just like i created only Auth Hooks here "Auth Hooks Use Postgres functions or HTTP endpoints to customize the behavior of Supabase Auth to meet your needs Docs All hooks Add hook Send Email hook Will be called by Supabase Auth each time an email message needs to be sent. type HTTPS endpoint endpoint https://zzmlainpucbkvcrtyrrg.supabase.co/functions/v1/create-profile-on-signup Copy secret \*\*\*\* \*\*\*\* \*\*\*\* \*\*\*\* Reveal Configure hook Docs"  
  
Question: Bro What Database indexes can do? Just like i created 3 Database indexes here "Database Indexes Improve query performance against your database Docs Index Advisor schema public Search for an index Create index Schema Table Name public profiles profiles\_pkey View definition public users users\_email\_key View definition public users users\_pkey View definition"  
  
Question: Bro What Database Triggers can do? Just like i created only Database trigger here "Database Triggers Execute a set of actions automatically on specified table events Docs schema auth Search for a trigger Viewing triggers from a protected schema The auth schema is managed by Supabase and is read-only through the dashboard. Learn more Name Table Function Events Orientation Enabled on\_auth\_user\_created users sync\_auth\_user\_to\_public AFTER INSERT ROW"  
  
Question: Bro What Authentication Policies can do? Just like i created only Authentication policies here "Policies Manage Row Level Security policies for your tables Docs Filter tables and policies schema public profiles Disable RLS Create policy Name Command Applied to Actions Enable insert for authenticated users only INSERT authenticated Enable read access for all users SELECT public Users can only access own profile ALL public users Disable RLS Create policy Name Command Applied to Actions Enable insert for all users INSERT public Users can view own data SELECT public"**