Real Time Messenger

NAME: SATVIK PRATAP SINGH
IIT KANPUR

System Design Documentation

Overview

- **Project Name:** FriendZone Dashboard
- **Description:** This system provides a user dashboard where users can manage their friends, group chats, and profile information. It integrates with Upstash Redis for handling real-time data like friend requests and chat sessions.
- **Primary Features:**
  - Adding friends
  - Viewing and managing chats with friends
  - Accessing group chats
  - Viewing friend requests and accepting/rejecting them
  - User profile with the ability to sign out
- **Tech Stack:**
  - **Frontend:** Next.js
  - **Backend:** Next.js (API routes), Upstash Redis for data storage and caching
  - **Authentication:** Next-Auth for managing user sessions and authentication

Architecture

- **Frontend (Next.js):**
  - The dashboard UI is rendered with React and server-side rendering through Next.js, which enables fast loading times and SEO-friendly pages.
  - **Components:** Icons, SidebarChatList, FriendRequestSidebarOptions, etc., to handle different sections of the dashboard.
- **Backend (Next.js API Routes):**
  - API routes handle session management (using `getServerSession` from Next-Auth) and fetching friends using helper functions like `getFriendsByUserId`.
  - **Redis (Upstash):** Upstash Redis is used for real-time operations such as fetching pending friend requests and managing active chat sessions.
- **Database (Upstash Redis):**

    - Redis serves as the data layer for storing friendship requests, chats, and real-time notifications.

Component Breakdown

- **Sidebar Components:** Handles navigation between adding friends, viewing chat lists, and accessing group chats. This includes UI logic and session management for showing personalized data.
- **Mobile Chat Layout:** A responsive design allowing mobile users to navigate and access their chats efficiently.
- **Helper Functions:**
    - `fetchRedis`: A utility to query Redis for friend requests or other session data.
    - `getFriendsByUserId`: Fetches a user's friends from Redis.

Data Flow

- **Authentication Flow:**
    - Users sign in via Next-Auth. The session is validated using `getServerSession` in the layout, redirecting them to the dashboard on success.
- **Friend Requests:**
    - Fetch pending friend requests from Upstash using `fetchRedis('smembers', ...)`.
- **Chat Management:**
    - The sidebar lists active chats using `SidebarChatList`, with chat data retrieved from Redis

Setup and Run Documentation

Prerequisites

Software:

- Node.js (v14+)
- npm or yarn
- Upstash Redis account (for real-time data storage)

Environment:

- Ensure the project is set up locally using a development environment like VSCode.
- You can also deploy the app to platforms like Vercel or Netlify, which support Next.js apps.

Installation and Setup

1.Clone the repository:

2.Install dependencies using

```
npm install
```

3.Set up environment variables: In your `.env.local` file, add the following variables:

```
NEXT_PUBLIC_UPSTASH_REDIS_REST_URL=<your-upstash-redis-url>
NEXT_PUBLIC_UPSTASH_REDIS_TOKEN=<your-upstash-token>
NEXTAUTH_URL=http://localhost:3000
```

(In case your own env file don’t work then i have attached my own .env local file.)

```
UPSTASH_REDIS_REST_URL=https://liberal-duckling-32773.upstash.io

UPSTASH_REDIS_REST_TOKEN=AYAFAAIjcDE3MDRjZTI3NDEzMzg0YzVmYWFlMDdjOTNhOTA1MzAwZXAxMA

GOOGLE_CLIENT_ID=644910345105-qba1duhu45p0ondrgfcd1ctcp6j7bg55.apps.googleusercontent.com

GOOGLE_CLIENT_SECRET=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

NEXTAUTH_SECRET=168181d9abc64bc53e75ac06d6624d4ca0972373f2da7d2ddb7bef97c58bcc83

PUSHER_APP_ID=1880109

NEXT_PUBLIC_PUSHER_APP_KEY=bd5dc99712548d7c6e64

PUSHER_APP_SECRET=efdf81eab4fdfa7f0fd2

PUSHER_CLUSTER=ap2
```

4.Running the Project using

npm run dev

Libraries and Dependencies and why we used them

- **Next.js:**
  - Provides server-side rendering and client-side interactivity for improved performance and user experience.
- **Upstash Redis:**
  - Chosen for its serverless nature and ability to handle high-performance real-time data, such as friend requests and chat sessions.
- **Next-Auth:**
  - Used for handling user authentication, making it easy to integrate with providers like Google or GitHub for login purposes.
- **Icons and Image Components:**
  - Icons are dynamically imported from a component folder, and `Image` from Next.js ensures optimized image loading and lazy loading.