Project Harmony - Mental Health Support Platform

Overview

Project Harmony is a web-based mental health platform that connects users with mental health professionals (psychologists, psychiatrists) and facilitates communication through video and voice calls. The platform also provides a forum for users to share their thoughts, experiences, and advice on mental health topics. The application integrates various technologies such as Next.js, Tailwind CSS, Framer Motion, Firebase Firestore, MongoDB, and Sendbird SDK to create a seamless, secure, and interactive experience for users.

Table of Contents

Technologies Used

Frontend:

- **Next.js**: Framework for building server-side rendered React applications.
- **Tailwind CSS**: A utility-first CSS framework for fast UI development.
- **Framer Motion**: A library for creating animations and interactive UI components.

Backend:

- **Firebase Firestore**: NoSQL cloud database for real-time data synchronization (e.g., chat messages).

- **MongoDB**: NoSQL database used for storing non-real-time data such as user profiles, forum posts, etc.
- **Sendbird SDK**: For implementing real-time video and voice call capabilities.

Others:

- **Firebase Authentication**: For user authentication and security.
- **Node.js**: Runtime environment for the backend.
- **Express.js**: Framework used for building APIs and server-side functionality.

Features

- **Real-Time Video and Voice Calls**: Video and voice calls between psychiatrists and clients using the Sendbird SDK.
- **Forum**: A space where users can share thoughts about their mental health and receive suggestions from others.
- **User Authentication**: Secure login and registration system powered by Firebase Authentication.
- **Real-Time Chat**: Messaging feature between users and professionals in real-time using Firebase Firestore.
- **Profile Management**: Ensures secure video and voice calls between psychiatrists and clients with encrypted communication.

Architecture Overview

- **Frontend**: Built with Next.js, which handles both server-side and client-side rendering. The UI is styled using Tailwind CSS, and interactive animations are powered by Framer Motion.
- **Backend**: The backend is built using Node.js with Express.js, managing authentication, routing, and integrating the Sendbird SDK.
- **Database**: Firebase Firestore handles real-time data (chat messages, online statuses), while MongoDB stores non-real-time data like forum posts, user profiles, and more.
- **Third-Party Integrations**:
  - Sendbird SDK for video and voice call services.
  - Firebase Authentication for secure and easy-to-integrate login.

Setup and Installation

Prerequisites

Before running the application locally, ensure that the following tools are installed:

- Node.js (version >= 14.x)
- npm or yarn
- MongoDB (for local development or use MongoDB Atlas)
- Firebase project with Firestore enabled
- Sendbird account and API key

Clone the Repository

bash

Copy

```
git clone https://github.com/your-username/project-harmony.git
cd project-harmony
```

Install Dependencies

bash

Copy

```
npm install
# or if using yarn
yarn install
```

Environment Variables

Create a .env.local file in the root of your project and add the following:

bash

Copy

```
GOOGLE_SECRET=your_google_secret_key
NEXTAUTH_URL=https://harmony-stoicbrains-github-io.vercel.app/
NEXT_PUBLIC_SECRET=your_public_secret_key
NEXT_PUBLIC_APP_ID=your_sendbird_app_id
MONGODB_URI=your_mongodb_connection_string
GOOGLE_ID=your_google_client_id
```

Run the Development Server

bash

Copy

```
npm run dev
```

or

yarn dev

Your app should now be running at http://localhost:3000.

Frontend Structure

The frontend is organized into several main sections:

- **Pages**: Contains React components corresponding to different routes of the app (e.g., index.js, forum.js, profile.js).
- **Components**: Reusable UI components such as buttons, forms, modals, and more.
- **Animations**: Uses Framer Motion for smooth page transitions and animations.
- **Styles**: Tailwind CSS configuration, custom utility classes, and global styles.

Backend Structure

The backend code is organized as follows:

- **Server**: The main entry point of the backend (server.js) sets up routes and initializes API endpoints.
- **Routes**: Contains all API routes, such as authentication routes, chat routes, and forum routes.
- **Controllers**: Handles the business logic for the routes.
- **Models**: Defines data models for MongoDB collections (e.g., users, forum posts, etc.).

API and Services

Firebase Authentication

Used for handling user registration, login, and session management. The API exposes routes like:

- POST /auth/signup: Registers a new user.
- POST /auth/login: Logs in an existing user.

Firebase Firestore

Used for handling real-time data synchronization for chat. The API exposes routes like:

- GET /chat/:userId: Fetches messages for a particular user.
- POST /chat: Sends a new chat message.

MongoDB

Handles non-real-time data such as forum posts, user profiles, and history. API routes include:

- GET /forum: Fetches all forum posts.
- POST /forum: Creates a new forum post.

Sendbird Integration

Connecting to Sendbird:

The app connects to the Sendbird server using an API key and generates unique user IDs for each session.

Call Features:

- **Voice and Video Calls**: Users can initiate and receive calls from psychiatrists via the Sendbird API.
- **Call History**: Call history is stored and can be viewed by users.

Database

Firebase Firestore

Used for real-time chat messages and online user statuses. It stores messages in a "messages" collection with fields like userId, message, and timestamp.

MongoDB

Used for storing user profiles, forum posts, and other non-real-time data. Collections:

- **Users**: Stores user data such as name, email, and role (client/psychiatrist).
- **Forum Posts**: Stores forum posts with fields like authorId, content, and timestamp.

Security Considerations

- **Authentication**: Firebase Authentication ensures secure login and session management.
- **Data Encryption**: Video and voice calls are encrypted end-to-end using Sendbird's built-in encryption.
- **Private Data**: Sensitive user data is stored securely, with access restricted using Firebase Firestore and MongoDB rules.

Future Enhancements

- **AI-based Recommendations**: Implement an AI-based recommendation engine to suggest mental health resources based on user preferences and behavior.

- **Multilingual Support**: Add support for multiple languages to make the platform accessible globally.
- **Mobile App**: Extend the platform to a mobile application using React Native or Flutter.

Contributing

Contributions are welcome! If you'd like to contribute to Project Harmony, please fork the repository, create a new branch, and submit a pull request. Make sure to follow the existing code style and write tests where necessary.

License

This project is licensed under the MIT License - see the LICENSE file for details.