TECH STACKS USED IN CHAT APP PROJECT

Next.js

- **What It Does:** Next.js is a powerful React framework for building server-rendered and statically generated applications. It simplifies routing, server-side rendering, and provides built-in optimizations.
- **Why I Used It:** I chose Next.js for its server-side rendering capabilities, which help improve performance and SEO. It also makes routing straightforward and handles server-side logic efficiently, which is crucial for a real-time application like mine.

TypeScript

- **What It Does:** TypeScript is a superset of JavaScript that adds static types. It helps in catching errors during development by providing type checking and offers better code refactoring and documentation.
- **Why I Used It:** I used TypeScript to enhance code quality and maintainability. Its type system helps prevent runtime errors and improves development efficiency by providing better tooling and autocomplete features.

MongoDB

-**What It Does:** MongoDB is a NoSQL database that stores data in a flexible, JSON-like format. It is designed for high performance and scalability, making it easy to handle large volumes of data.
-**Why I Used It:** MongoDB is ideal for storing chat data and user profiles because of its scalability and flexibility. It handles various data types and structures well, which suits the needs of a real-time messaging application.

Prisma

-**What It Does**: Prisma is an ORM tool that offers type-safe database queries and schema management. It simplifies interactions with the database and integrates well with various databases.
- **Why I Used It**: I chose Prisma to manage and query MongoDB data more easily and safely. Its type-safe querying reduces the likelihood of bugs and enhances the development workflow.

Socket.IO

- **What It Does**: Socket.IO is a library for real-time, bidirectional communication between clients and servers. It supports WebSockets and other transport protocols for real-time messaging.
- **Why I Used It**: Socket.IO is essential for enabling real-time messaging in my chat application. It ensures that messages are delivered instantly and updates are communicated in real-time.

Pusher

- **What It Does:** Pusher provides real-time messaging and event broadcasting services. It simplifies the implementation of real-time features and manages the infrastructure needed for real-time interactions.
- **Why I Used It:** I used Pusher to handle real-time updates and event broadcasting in my chat app. It complements Socket.IO by providing additional capabilities for scaling and managing real-time features.

Cloudinary

- **What It Does:** Cloudinary is a cloud service for managing and delivering images and videos. It offers features for image uploading, storage, transformation, and optimization.
- **Why I Used It:** I integrated Cloudinary to handle image sharing in the chat application. It allows users to upload and share images efficiently while optimizing them for better performance and delivery.