

Design-Report

I. Database

Overview

Our database is designed to store user information, including usernames and passwords. It also stores messages and replies to those messages, along with the users who posted them and timestamps of when they were created.

Schema

Our database has three main tables: users, messages, and replies.

Relationships

There are several relationships between the tables in our database:

A user can post many messages, but a message can only be posted by one user (one-to-many relationship between users and messages)

A message can have many replies, but a reply can only be in response to one message (one-to-many relationship between messages and replies)

A user can post many replies, and a reply can only be posted by one user (one-to-many relationship between users and replies)

Architecture

Our database is a MySQL database hosted on a remote server. We connect to the database using a Node.js backend, which handles all of the database queries and sends data to the frontend.

II. React Application

Overview

Our React application is a simple messaging app that allows users to create accounts, post messages, and reply to messages. It has a responsive design and is optimized for mobile and desktop devices.

Architecture

Our React application is structured using the following components:

App: The main component that contains all other components

Header: The header component that contains the logo and the sign in/sign out button

Auth: The sign in/sign out component that allows users to create accounts and log in or log out

Messages: The component that displays all messages and allows users to post new messages

Message: The individual message component that displays the message content, username, and replies

NewReplyForm: The component that allows users to post new replies to messages

We use react-router-dom to handle navigation between different components. We also use axios to make API calls to the Node.js backend to fetch and post data to the database.

Styling

We use styled-components to style our React components. Our design is inspired by popular messaging apps and uses a clean and modern color scheme. We use media queries to ensure our design is responsive and looks good on all devices.