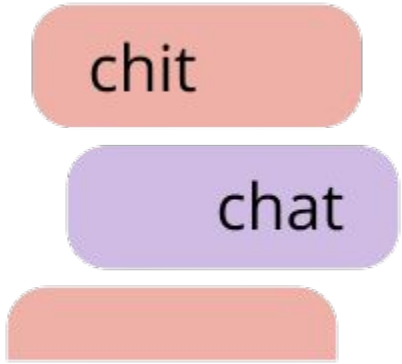


A web app by
Colin Snow, Maia Materman, and Shirin Kuppusamy

Introduction

Architecture and Challenges

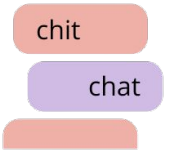
Demo

What We Learned

chit

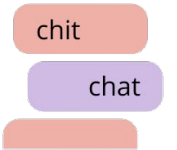
chat

INTRODUCTION



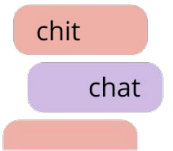
Motivation

There are a billion chat apps on the market right now. Why not make one more?



Overall Goals

We wanted to build a beautiful website that had nontrivial frontend, backend, and database components. We also wanted to ensure that we wrote a chat app that actually worked!

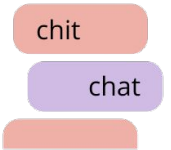


Learning Goals

Maia: I hoped to improve my React skills and really understand and execute on design and front-end development!

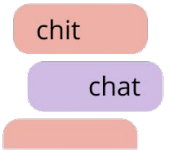
Shirin: I wanted to learn more about SQL and work with information on the backend to create a coherent product!

Colin: I hoped to learn about sockets and frontend/backend communication.



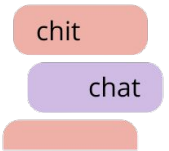
Features

- Synchronous communications between users and continuously updating pages
- Branded and aesthetically pleasing (at least we think!)
- Comprehensive error checking

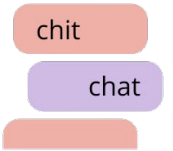


Functions

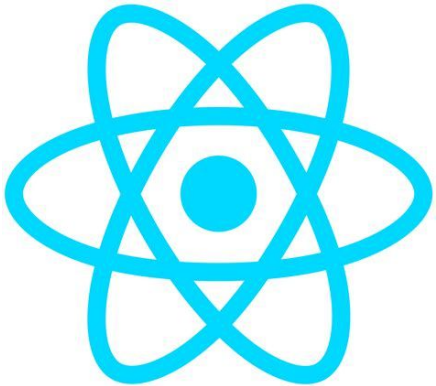
- Create new user accounts
- Update user profile information
- Delete user accounts
- Render and open all chats and messages
- Write new messages in real time
- Login to multiple accounts at same time



ARCHITECTURE AND CHALLENGES



Technology Stack



chit

chat

Our Designs: Login and Profile

Maia Materman

BACK

Name: Maia Materman EDIT

Username: webdevchamp23 EDIT

Password: ********** CHANGE

DELETE MY ACCOUNT

Create Account

Username

Password

LOGIN

Forgot Password?

chit

chat

Our Designs: Create Account and Forgot Password

BACK

Name

Username

Password

Confirm Password

CREATE ACCOUNT

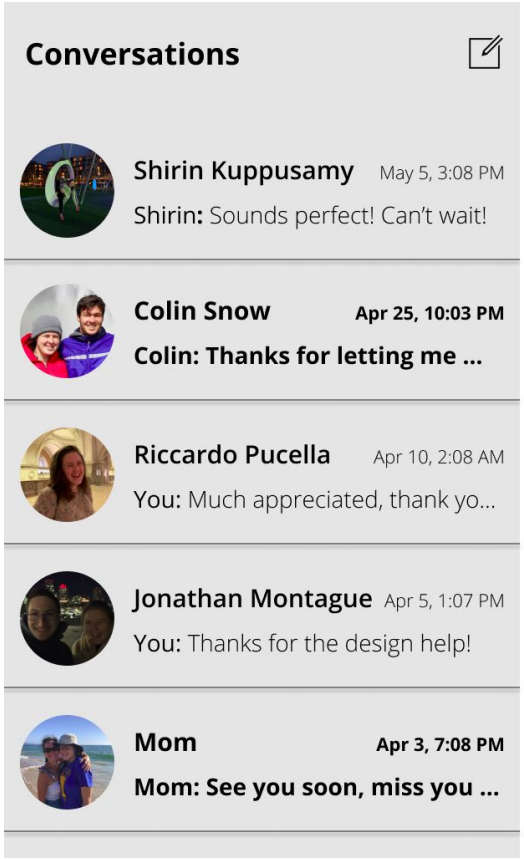
BACK

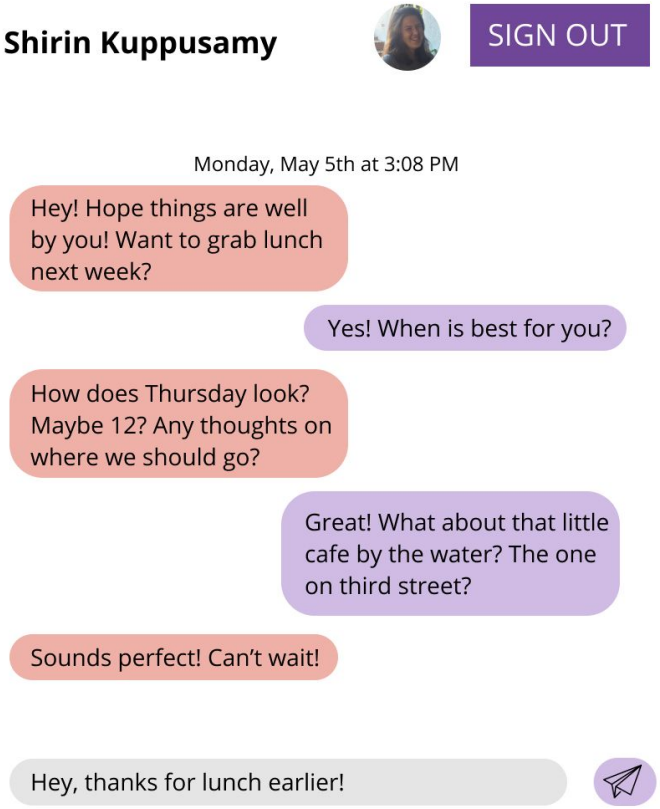
Sorry, that's too bad.

chit

chat

Our Designs: Main Chat Page



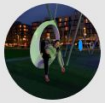


chit

chat

Our Designs: New Message

Conversations



Shirin Kuppusamy May 5, 3:08 PM

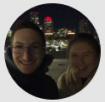
Shirin: Sounds perfect! Can't wait!

Colin Snow **Apr 25, 10:03 PM**

Colin: Thanks for letting me ...

Riccardo Pucella Apr 10, 2:08 AM

You: Much appreciated, thank yo...



Jonathan Montague Apr 5, 1:07 PM

You: Thanks for the design help!

Mom **Apr 3, 7:08 PM**

Mom: See you soon, miss you ...

To: Sh

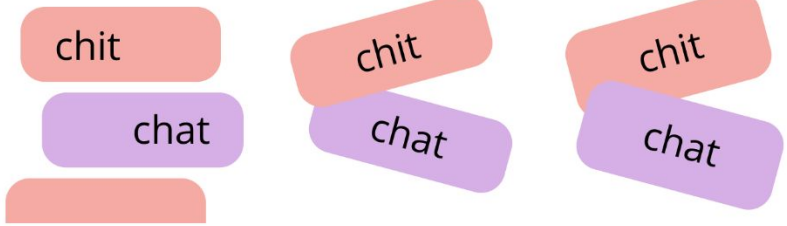
SIGN OUT

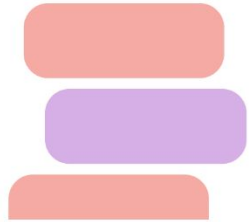
Hello!

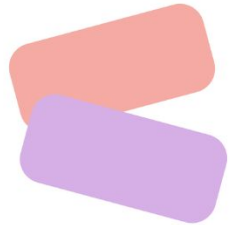
chit

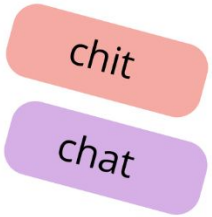
chat

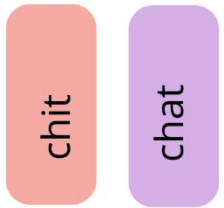
Our Designs: Logo

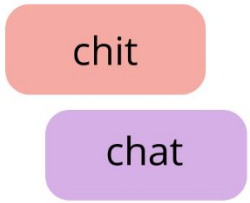


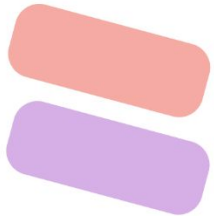


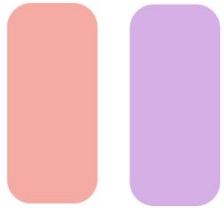


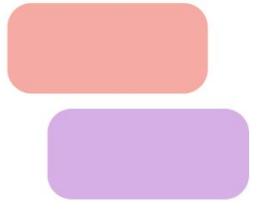


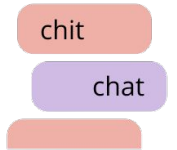




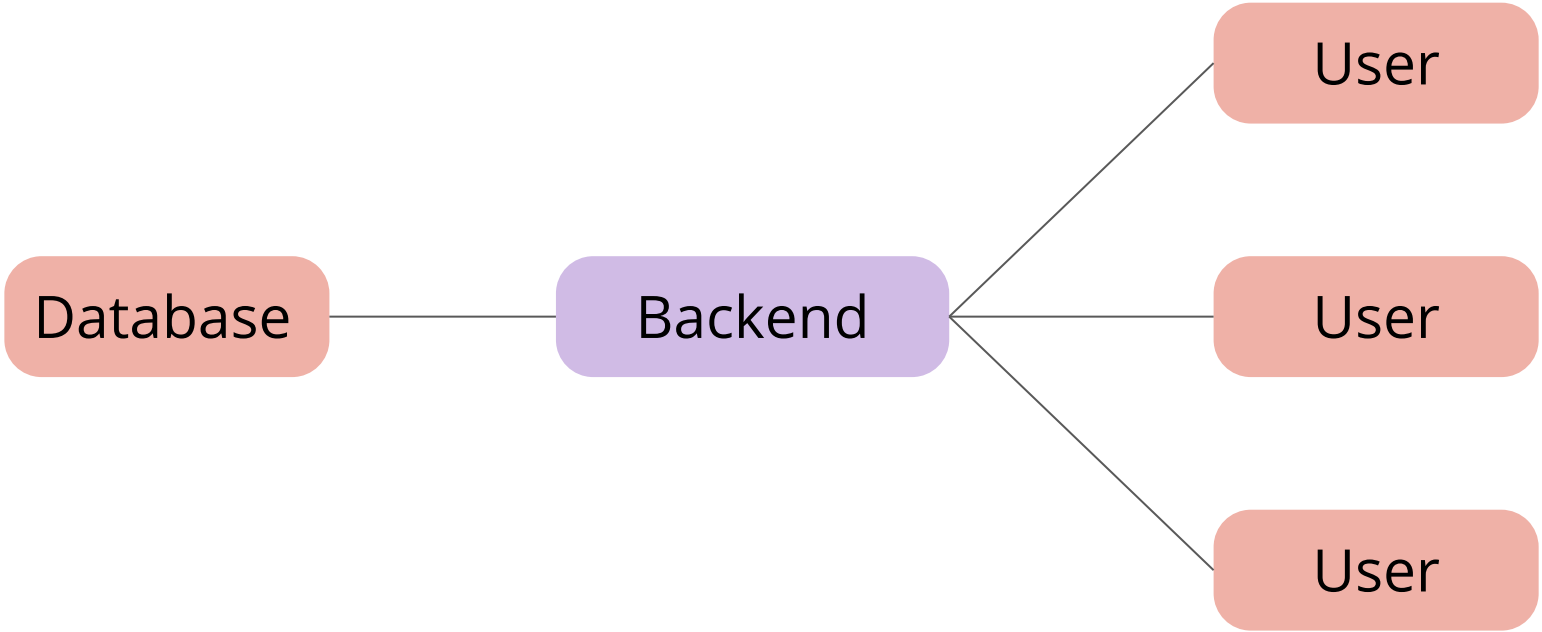








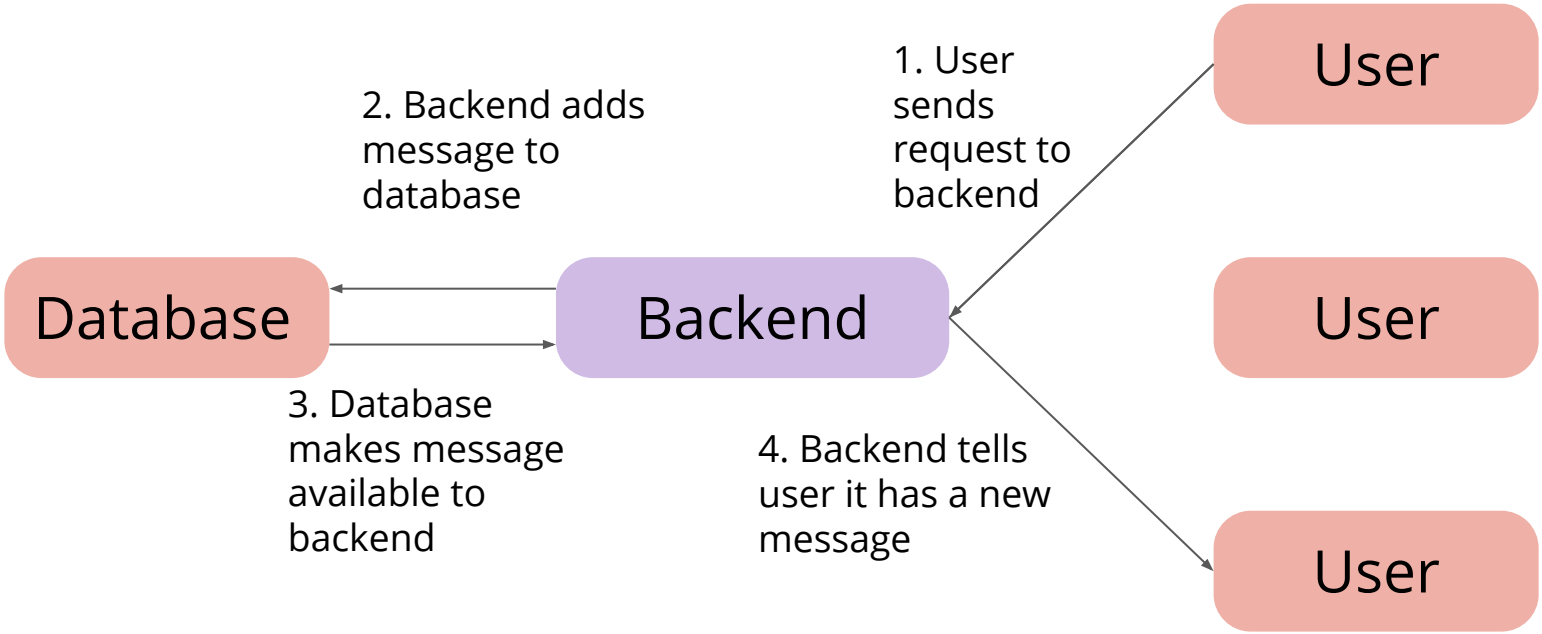
System Diagram



chit

chat

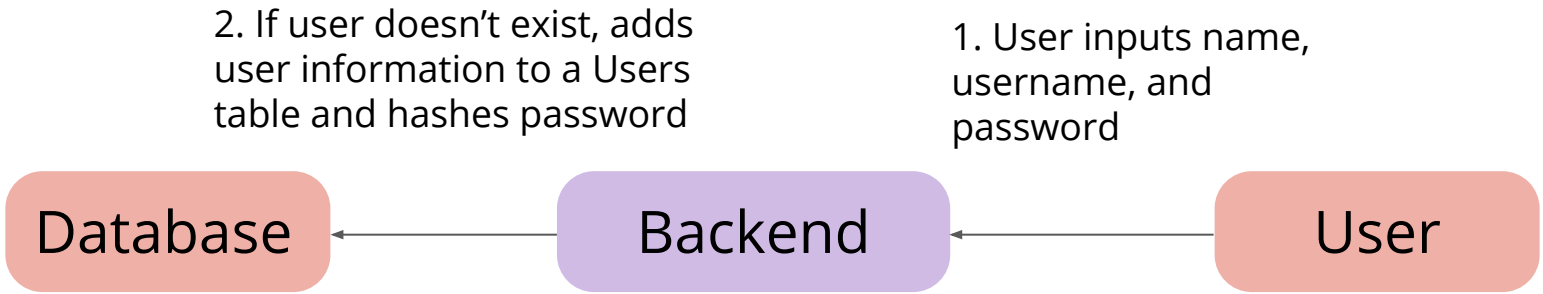
New Message Process



chit

chat

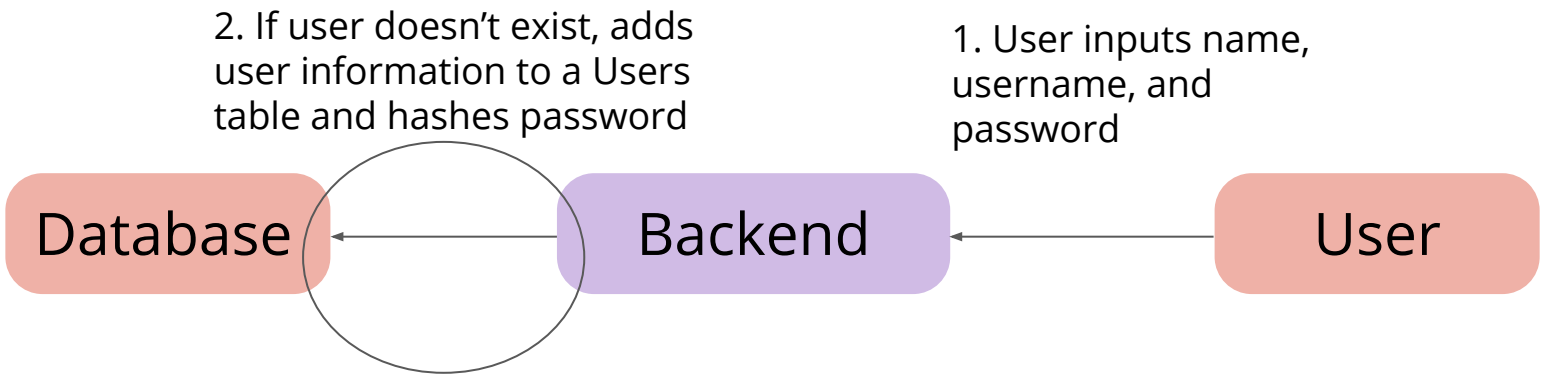
New User Creation Process



chit

chat

New User Creation Process

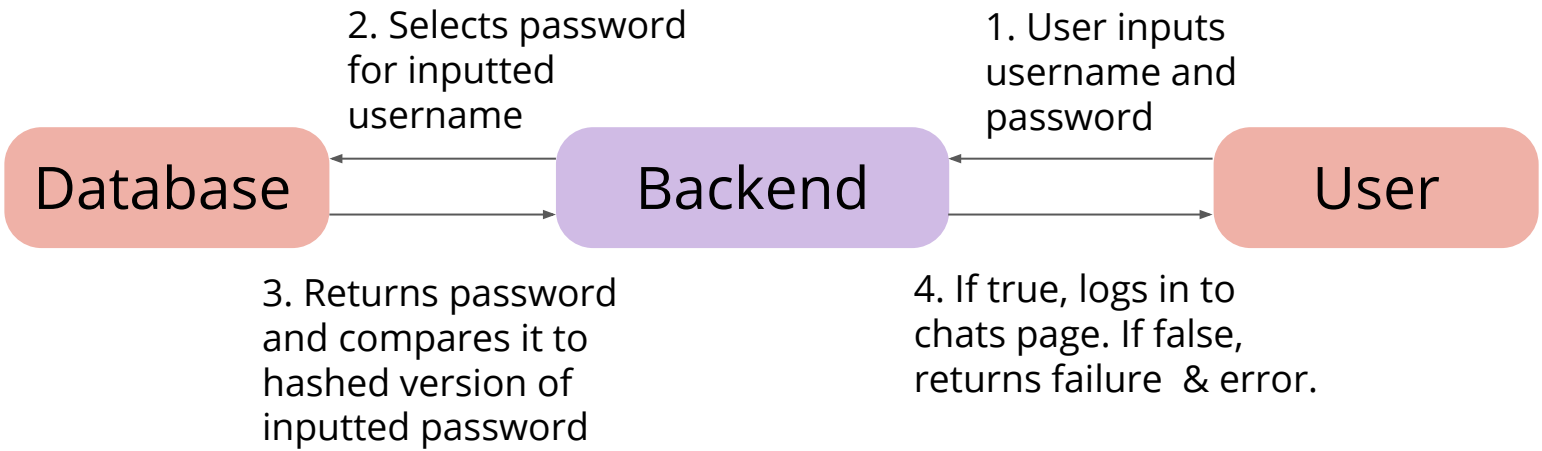


Let’s zoom in on checking if the username exists!

chit

chat

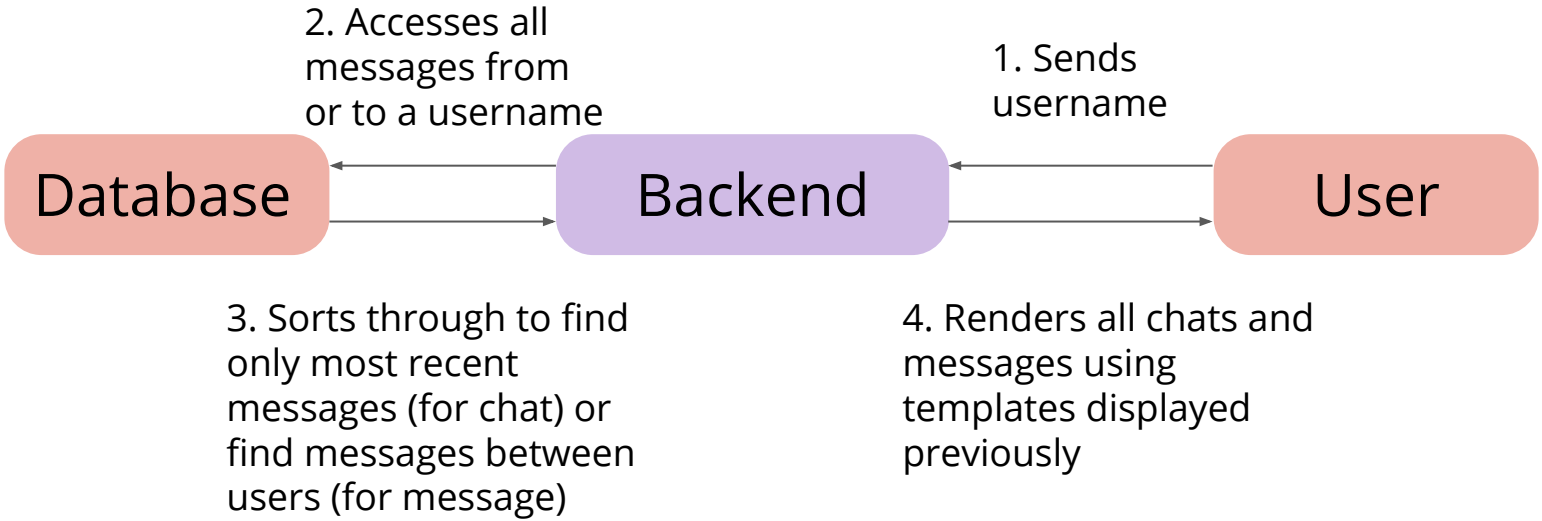
Login Process



chit

chat

Get Message/Chat Process

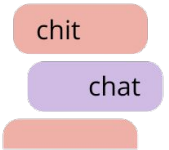


chit

chat

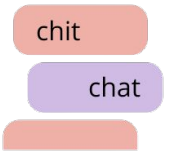
Challenge 1: Real Time Communication

- HTTP protocol was never meant for real-time communication
- A server usually only responds when requested
- Frontend would have to ask constantly for messages



Solution: Websockets

- More modern protocol, defined in 2011
- Enables full two-way communication
- Either client or server can send a message to be read by the other



Solution: Websockets

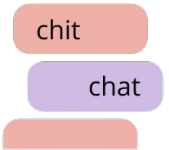
Frontend:

```
socket.emit("login", {data})
```

```
socket.on("login_response", data => {
      ...
});
```

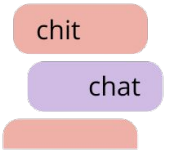
Backend:

```
@socketio.on('login')
def login(data):
    ...
    emit("login_response", data)
```



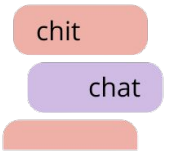
Challenge 2: Dynamic Rendering

- Need to handle everything from new user with no chats to users with many chats and messages
- Page has to update every time there is a new message without obviously reloading



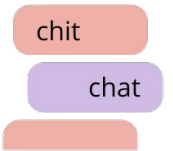
Solution: React and Conditional Rendering

- Update only components which change
- Use state to signal user actions



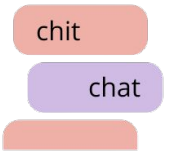
Challenge 3: Database Communication

- Need to ensure thread-safety
- Need to find users quickly
- Data size gets large



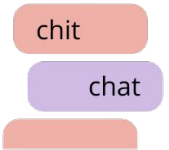
Solution: MySQL

- Guaranteed thread safe
- Can store users and messages in structured form
- Sort messages with username as primary key



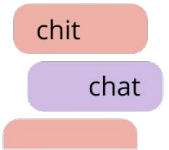
Challenge 3: Database Communication

- The syntax of using mySQL statements in a python script has nuances that can be hard to pinpoint
- Making sure the emit is different from the route is important or routes are called repeatedly
- Handling large amounts of data



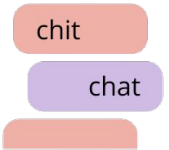
Solution: Database Communication

- Asking for help from more experienced SQL users!!
- Making sure to use unique, understandable, and consistent naming conventions
- Using multiple tables and primary keys to keep data sortable and accessible



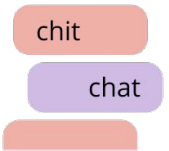
Challenge 4: Ensuring Security

- Passwords stored in a database can be a major security hazard
- A user can input a username that another user already has

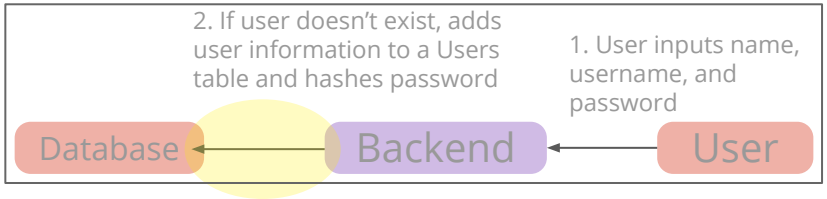


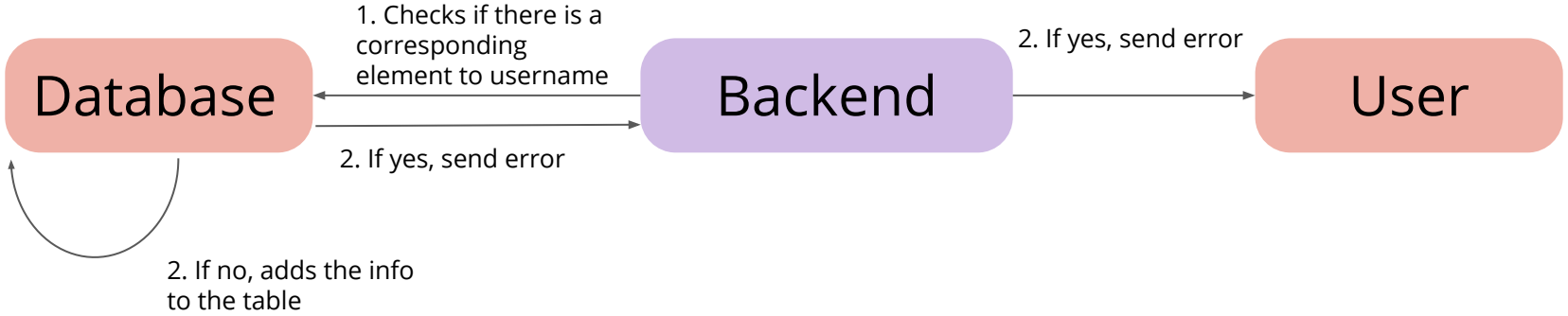
Solution: Ensuring Security

- md5 hashing for passwords
- Checks to see if user exists
- Reroute to login on username change



User Checking Process



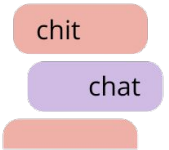


chit

chat

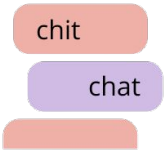
Challenge 5: Testing Edge Cases

- There are many ways a user can interact with our interface
- Sometimes a user may try to incorrectly interact with the interface
- Many errors more than what are useful for the user are thrown

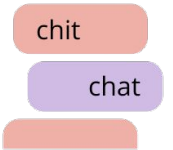


Challenge 5: Testing Edge Cases

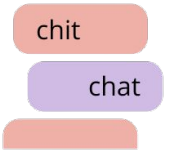
- We did a full run-through of our chat server several times
- We logged in with multiple users at different “stages” of messaging
- Flagging errors that are useful to the user and creating pop-ups



DEMO

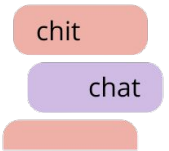


WHAT WE LEARNED



Insight 1: New languages

- Learned a lot of new terminology in order to communicate with one another and ask for help
- New languages are hard to pick up and use in a few weeks!
- Writing code and writing efficient code are two different things, and sometimes you make trade-offs



Insight 2: Siloed Work/Overscoped

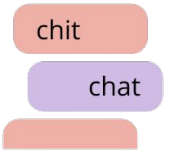
- We took on a lot of work in a lot of different fields
- To get it done, all needed to be independent and work on own time
- Being clear about what we want and what we are working on
- Integrating often and thoroughly

chit

chat

Insight 3: Little things, big impacts

- Small inconsistencies in our CSS caused big, hard to debug issues
- Minor issues with ordering in the database or syntax errors can cause long chains of error messages
- Once the database and socket were integrated, hard to test one without the other, so lot of “test case” development either in SQL server or sockettest file



QUESTIONS?