

ChatterBox

Samuel Tuki, Jonathan Davis,
Toyosi Akindele, Hui Zhao





Elevator Pitch

- A lightweight, WhatsApp like chat app for the world!
- Find new friends
- Chat about what you're interested in
- Take part in the global conversation
- Cure boredom
- Just hang out and watch the messages roll by
- Get answers to that niggling question

Table of Contents

1

What we did

2

Why we did it

3

How we achieved it

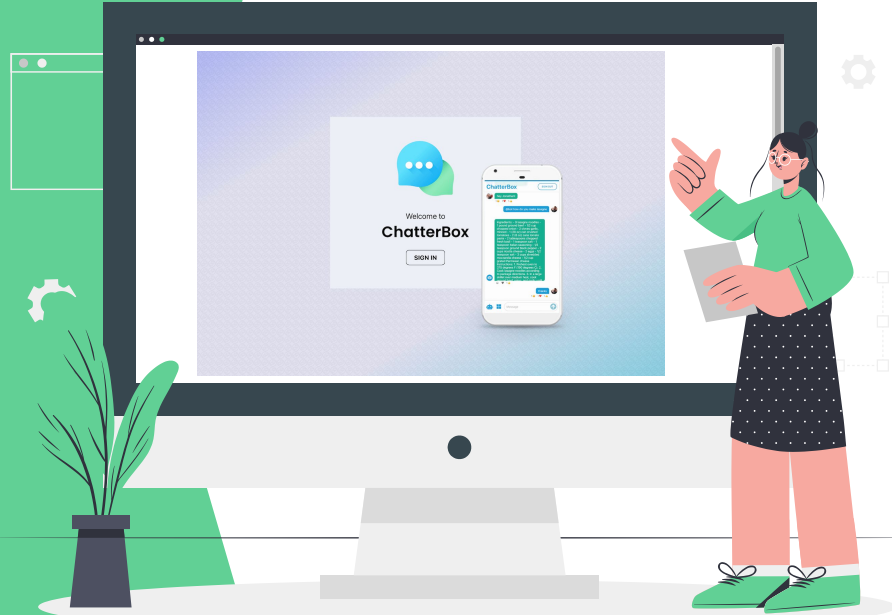
4

Potential developments

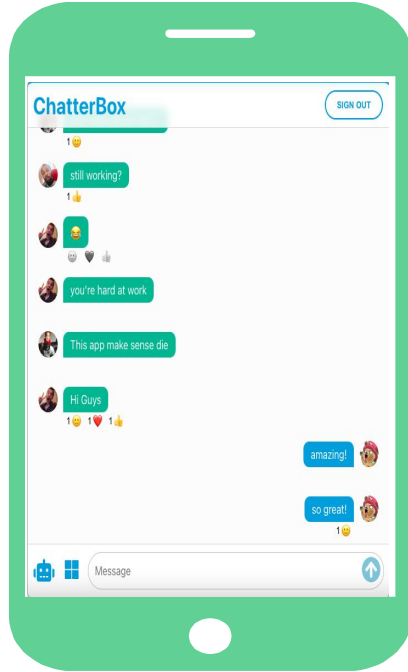
1

First Section

What we did
Why we did it



Our motivation for development



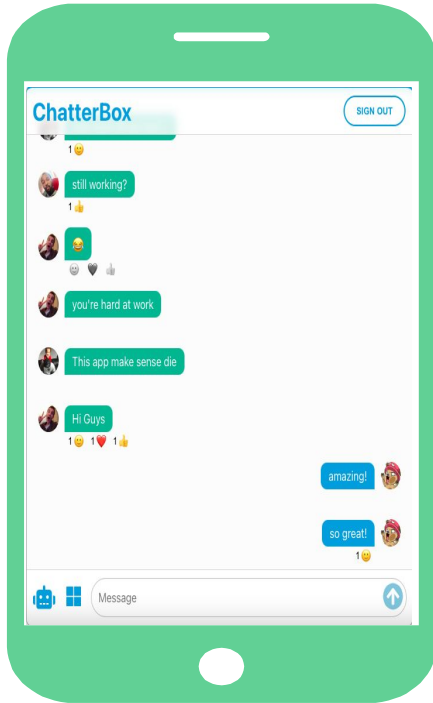
We wanted to implement a backend that would update the UI live

Chat app was a natural fit

Manageable for 4 people in 2 weeks

Chat with the world about everything, everywhere, all at once!

User story



As a user

I want to have a lightweight online chat software.

I want to have a chatbot that I can ask questions and get an answer.

I want to interact with other messages, give them likes, thumbs up, etc.

I also want to have a breakout rooms, so I can have more private chats or conversations about a particular topic.

SO THAT I can chat with others about anything and everything.

Acceptance Criteria.

- Deployed to GitHub Pages.
- Must use ReactJS.
- Must use Node.
- Must have both GET and POST routes for retrieving and adding new data.
- Must deploy this application using Netlify.
- Must utilize at least two libraries, packages, or technologies that we haven't discussed.
- Must have a polished front end/UI.
- Must meet good quality coding standards (indentation, scoping, naming).
- Repository meets quality coding standards with a quality README.

Acceptance Criteria.

It will be done when:

- Users can log in through their google account.
- Users can chat with each other in the chatroom.
- Users can ask a chatbot anything they want to know.
- Users can create or join a breakout room to have their own conversations.
- Users can give other messages likes, thumbs up and laughs.
- Users can delete their own message.

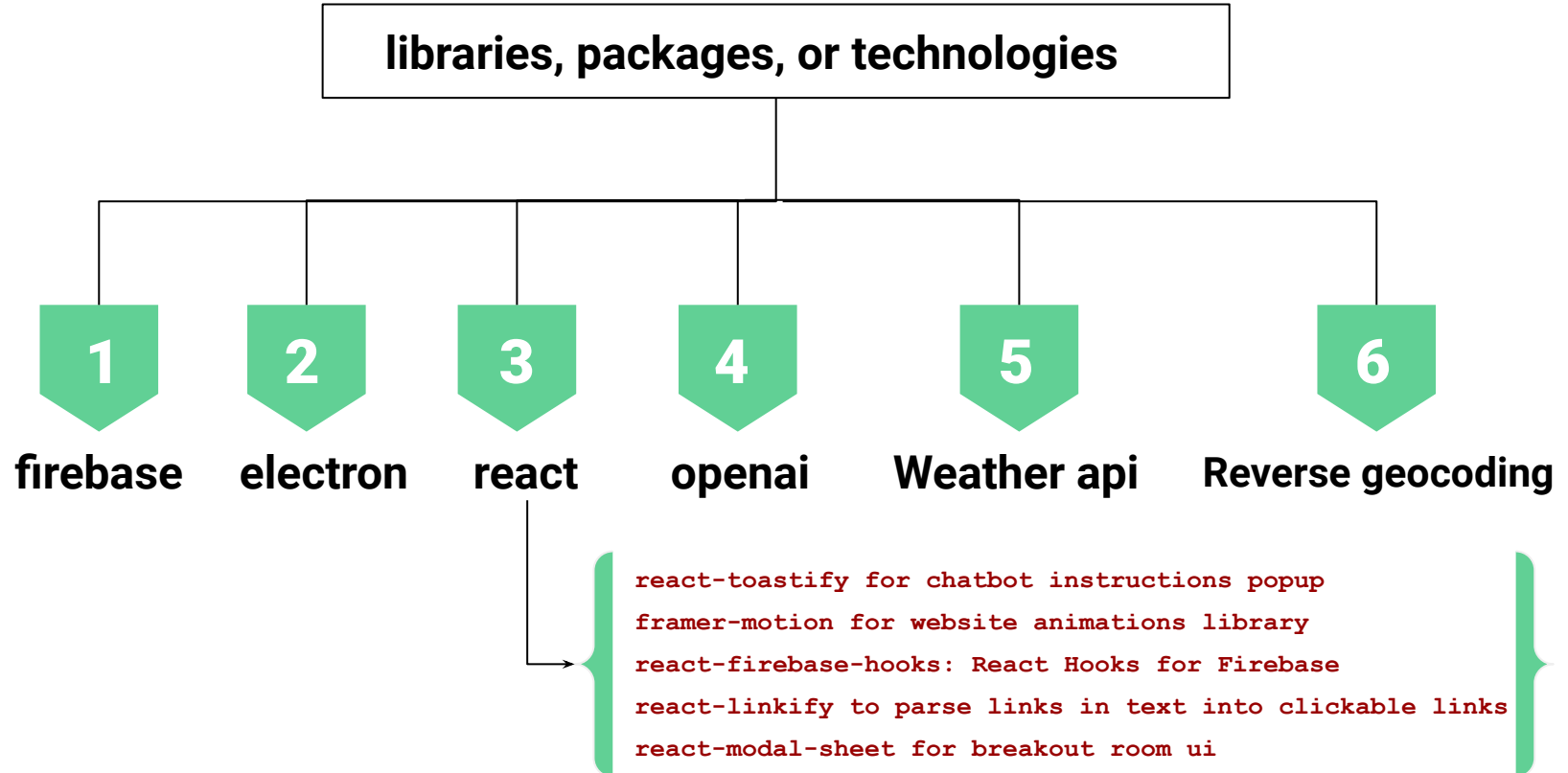
2

Second Section

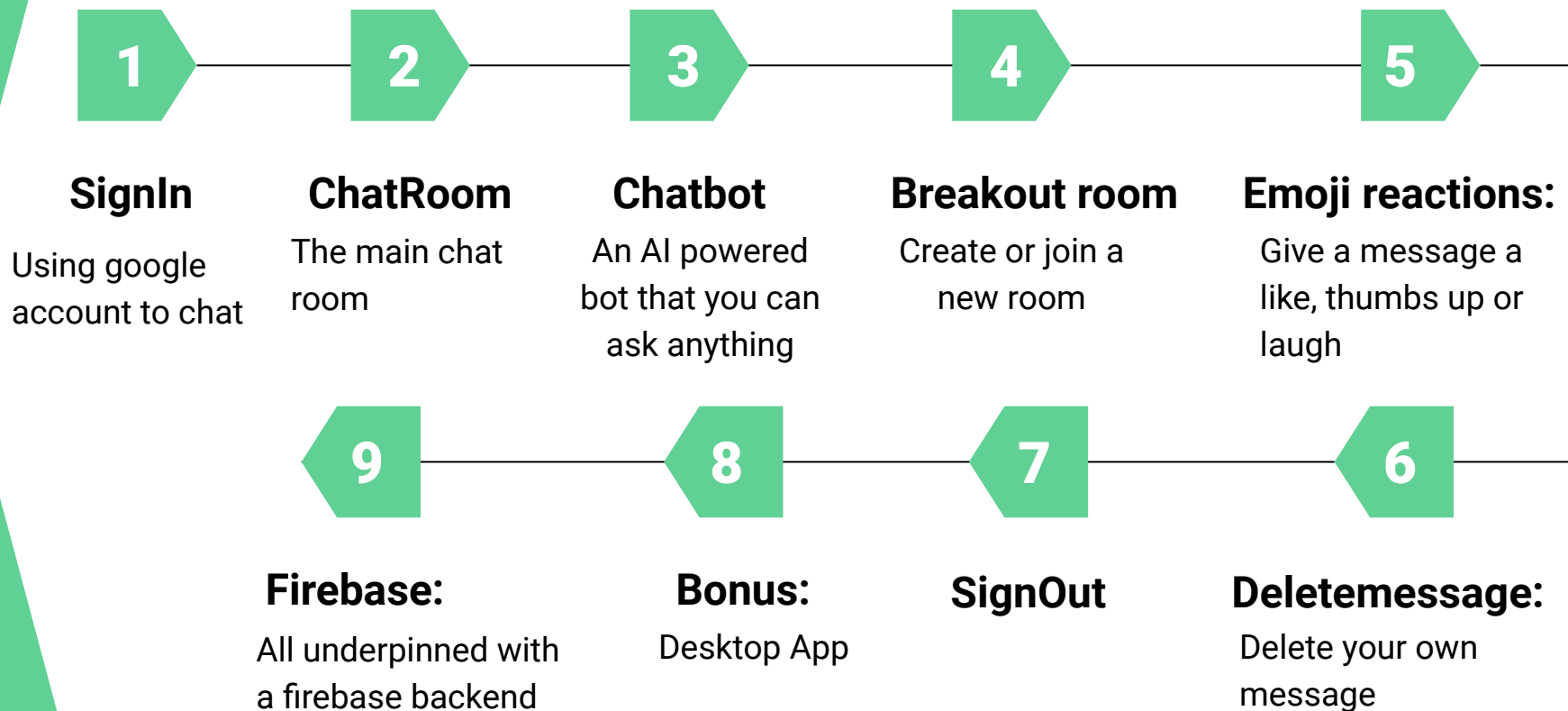
How we achieved it



Main technologies used



Breakdown of tasks and roles



Breakdown of tasks and roles

Samuel

- Project management
- Landing page
- Breakout rooms
- Breakout rooms modal
- Style revamp

Breakdown of tasks and roles

Jonathan

- Hooking app up to firebase/firestore
- User sign-in / authentication
- Message send / receive / delete
- Initial app styling & logic
- Chatbot

Breakdown of tasks and roles

Victor

- Message delete interface
- Emoji reactions
- Animation effects
- Sound effects
- Style revamp
- Securing API keys
- Chatbot usage info

Breakdown of tasks and roles

Hui Zhao

- Readme
- Presentation
- User story
- Acceptance criteria
- Laughing emoji reaction
- Bonus parts
 - Try to migrate from web to desktop
 - Use electron

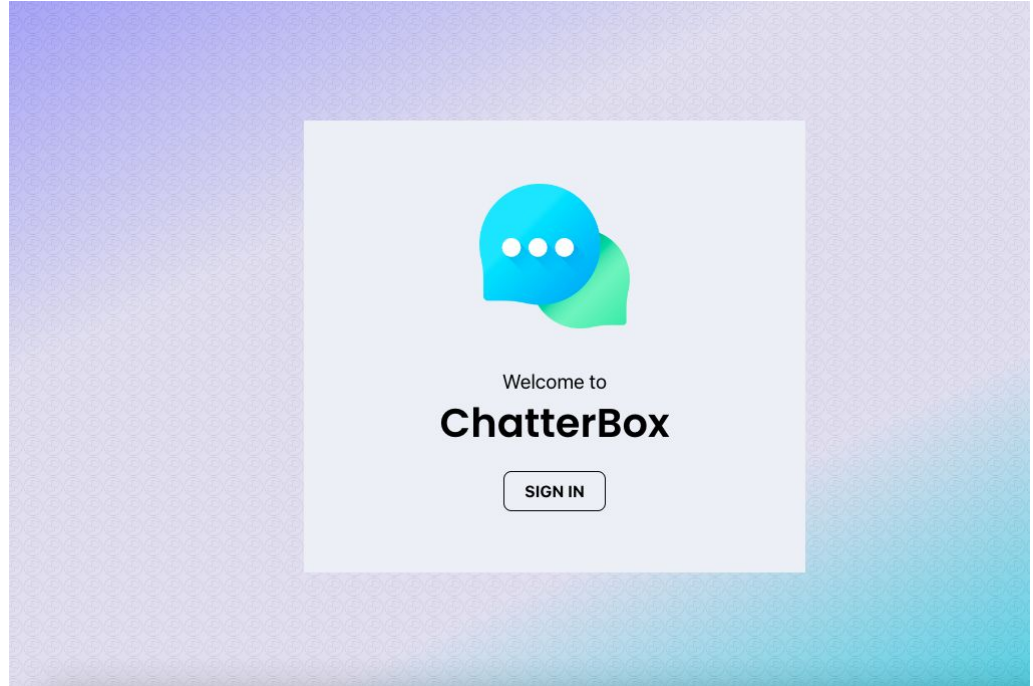
Demo

- **Github URL:**

<https://github.com/trunten/ubbc-chatterbox>

- **Deployed Site:**

chatterbox-web.netlify.app



Challenges & Achievements

Challenges

- Chat room anchoring
- Emoji reactions - add/remove
- Sound effects

Achievements

- Creating a chat app!!
- Implementing a backend that updates live (firebase/websockets)
- Breakout rooms for private chat
- Message emoji reactions
- Desktop application

Directions for Future Development

- Custom emoji reactions
- Message edits
- Breakout room directory
- npm package - `react-firechat`!
- Profanity filter (or use the perspective api to get AI filtering of harmful/toxic comments)

Thanks!

