Remote Theater

Hupaul Camacho, Chukwuka Okonkwo, Kadijah Wilson

Hupaul Camacho

/hupaulcamacho

/hupaulcamacho

Chukwuka Okonkwo

/Iakerfan1994

/Chukwuka-Okonkwo

Kadijah Wilson

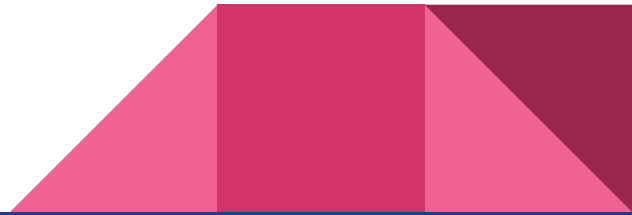
/KadijahW

/Kadijah-Wilson

The Remote Theater Dev Team

Social Distancing

- Movie theaters are closed, preventing people from having a physical movie watching experience
- Other similar movie watching platforms require a person to be in control of the media playing



An Online Movie Theater Application

Using Remote Theater, users can:

- **Watch** movies playing at scheduled times
- **Chat** and **interact** with other users in real-time
- **Choose** preferences to get a personalized list of movies

App Demo & Tech Milestones

Syncing Viewers

Viewers all watch the same movie at the same time

```
setTime = () => {
    const { routeprops: { match:{ params } } } = this.props;
    const time = params.time
    return time
}

getTimeStamp = () => {
    const format = 'h:mm:ss A '
    let time = this.setTime()
    let showtime = moment(time, format)
    let now = moment().format(format)
    let duration = moment.duration(moment().diff(showtime))
    return duration.as('seconds')
}

_onReady = (event) => {
    let timestamp = this.getTimeStamp()
    console.log(timestamp, "timestamp")
    event.target.playVideo();
    event.target.seekTo(timestamp);
}
```

Real-Time Chat

Live chat tailored for interactions and discussions between viewers

```
componentDidMount = async () => {

  const { user, title } = this.state
  const movietitle = title.split(":").join("")
  const movietitle2 = movietitle.split(" ").join("")
  const chatClient = new StreamChat('ewnpysxxpud8');
  let response = await axios.post(`/api/getToken`, user);
  let token = response.data.token;
  let chatUser = response.data.user

  await chatClient.setUser(
    {
      id: chatUser.name,
      name: chatUser.name,
      image: `https://getstream.io/random_svg/?name=${chatUser.name}`
    },
    token,
  );
  let channel = chatClient.channel('livestream', movietitle2, {
    image: 'https://goo.gl/Zefkbx',
    name: 'test',
  });

  await channel.delete();

  this.setState({
    chatClient: chatClient,
    channel: channel
  })

}
```

Database

PostgreSQL or MongoDB

```
CREATE TABLE users
(
    id SERIAL PRIMARY KEY,
    name VARCHAR,
    email VARCHAR NOT NULL UNIQUE,
    number VARCHAR,
    password VARCHAR NOT NULL,
    video_id INT REFERENCES videos(id) on delete cascade on update cascade
);

CREATE TABLE showrooms
(
    id SERIAL PRIMARY KEY,
    title VARCHAR,
    video_id INT REFERENCES videos(id) on delete cascade on update cascade
);

CREATE TABLE comments
(
    id SERIAL PRIMARY KEY ,
    body VARCHAR,
    users_id INT REFERENCES users(id) on delete cascade on update cascade,
    video_id INT REFERENCES videos(id) on delete cascade on update cascade,
    showroom_id INT REFERENCES showrooms(id) on delete cascade on update cascade
);

CREATE TABLE preferences
(
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(id),
    genre_id INT REFERENCES genres(id)
);
```

Future Implementations

- Private theater feature that allows users to share a link and host movies for small groups
- A search feature where users can search for movies by name and genre
- A rating feature that influences recommended movie lists
- Add Shows and episodes so there’s more of a variety for users to watch

Judges Q&A