Final Project: We.TV

User Stories

1. User

- As a user I would like to register.
- I would like a profile with a picture(stretch), first name, last name, display name, email, and password.
- I would like to login.
- I would like to see my dashboard.
- My dashboard should have my shows, relevant posts.
- I would like to search for existing shows.
- My search returns a show, I would like to visit the page and join the group.
- I would like to view all posts by other users on the page.
- I would like to “like” a post.(toggle)
- I would like to “dislike” a post.(toggle)
- I would like to reply to these posts.
- I would like to create posts on my shows. Upload images, GIFs and videos.
- I would like to create polls on these pages.
- I would like to edit/delete my own posts.
- I would like to have a live chat with other users on the show page.
- I should be able to leave a group.
- I would like to log out.

ROUTES

BACKEND

Users

GET	/api/users	get all users
GET	/api/users/:user_id	get one user
POST	/api/users/register	create a user
PUT	/api/users/:user_id	update a user
POST	/api/users/login	login
POST	/api/users/authenticate	authenticate

JOIN TABLES

GET	/api/users/:user_id/shows	get all favourited shows for one user

Shows

GET	/api/shows	get all shows
GET	/api/shows/:show_id	get one show
POST	/api/shows/new	create a show

Posts

GET	/api/posts	get all posts
GET	/api/posts/:show_id	get all posts for one show
GET	/api/posts/:user_id	get all posts created by one user
POST	/api/posts	create a post
PUT	/api/posts/:post_id	update a post (send an object with update options)

DELETE	/api/posts/:post_id	delete a post

Comments

GET	/api/comments	get all comments
GET	/api/comments/:post_id	get all comments for one post
POST	/api/comments	create a comment
PUT	/api/comments/:comment_id	update a comment
DELETE	/api/comments/:comment_id	delete a comment

Chat Room Messages

GET	/api/chat_room_messages	get all messages
GET	/api/chat_room_messages/:chat_room_message_id	get a single message
GET	/api/chat_room_messages/:chat_room_id	get all messages for a chat room
GET	/api/chat_room_messages/:chat_room_id/:creator_id	get all messages for a single user in a chat room
POST	/api/chat_room_messages	create a message

Favourites

GET	/api/favourites	get all favourites
GET	/api/:user/favourites	get all favourites for one user
POST	/api/:user/favourites	create a favourite for a user

FRONTEND

/login	login page
/register	register page
/user/shows	my shows page
/user/posts	my posts
/profile	my profile
/search	search page
/shows/:show_id	a specific show page

REACT COMPONENTS

HOME

<TV>(with static)
<Login_button>
<Register_button>

LOGIN

<Form>
    <Email>
    <Password>
    <Submit>

REGISTER

<Form>
    <Name>
    <Display_Name>
    <Email>
    <Password>
    <Confirm_Password>
    <Submit>

NAV BAR

<My_Shows>
<My_Posts>
<My_Profile>
<Search>
<Logout>

MY SHOWS

<Search_Shows_Filter>(update state, which is being passed to Shows Component)

<Shows>

<Show_card>

<Empty_show_card>

MY POSTS

<Posts>

<Post>

<Text_display>

<Image_display>

<Video_display>

<Poll_display>

<?>

<Likes>

<Dislikes>

<Add_comment>

<Add_comment_input>

<Text_Area>

<Image_Upload>

<Video_Upload>

<Submit>

<Comment>(Same component as Post)

<Text_display>

<Image_display>

<Video_display>

<Likes>

<Dislikes>

MY PROFILE

<Image>

<Edit_Image>

<Edit_Info>

<My_Info_Form>

<Name>

<Email>

<Display_name>

<Submit>

SEARCH

<Search_Input>

<Results>(Same as show card)

SHOW CHAT

<Show>
    <ShowBanner>
    <Show_Nav>
        <View_Chat>
        <View_Posts>
    <Chat props={chat with the show_id, onSubmit}>
        <ChatMessage props={message, timestamp}>
        <ChatForm props={props.onSubmit}>

SHOW POSTS

<Show>
    <Follow>
    <ShowBanner>
        <Show Info?>
    <Show_Nav>
        <View_Chat>
        <View_Posts>
    <Search_Post_Filter>(update state, which is being passed to Posts Component)
    <Add_Post>
        <Text_Area>
        <Image_Upload>
        <Video_Upload>
        <Poll_Upload>
        <Submit>
    <PostsList>
        <Post>
            <Text_display>
            <Image_display>
            <Video_display>
            <Poll_display>
                <?>
            <Likes>
            <Dislikes>
            <Add_comment>
            <Add_comment_input>
                <Text_Area>
                <Image_Upload>
                <Video_Upload>
                <Submit>
            <Comment>(Same component as Post)

<Text_display>
<Image_display>
<Video_display>
<Likes>
<Dislikes>
<Add_Post Button>

REACT STATES

Visual mode state(we can borrow state from scheduler and build on that)(changed by nav bar buttons)

Shows would work better as routes than states. Because we now have something in our database for each show.

That show page will it have a state?

Search area would have two states. One state where we're searching. One where we've searched in which case it would display the show page.

[click one of the show cards, the imdb_id for that card gets stored in a current_show state, the visual_mode state is set to null, we get show info for api using this state and get posts from local database using the state]

Another state for show page. One is chat one is posts.

Material UI Potentially useful Components

https://material-ui.com/components/bottom-navigation/
https://material-ui.com/components/drawers/
https://material-ui.com/components/menus/
https://material-ui.com/components/tabs/
https://material-ui.com/components/app-bar/
https://material-ui.com/components/cards/
https://material-ui.com/components/progress/
Backdrop React Component
https://material-ui.com/components/avatars/
https://material-ui.com/components/icons/
https://material-ui.com/components/material-icons/
material-ui/core/styles vs @material-ui/styles
https://material-ui.com/system/positions/
https://material-ui.com/system/flexbox/

https://material-ui.com/components/click-away-listener/
https://material-ui.com/components/textarea-autosize/
https://material-ui.com/components/autocomplete/
https://material-ui.com/components/skeleton/

Potential useful them changer

https://codesandbox.io/s/intelligent-snowflake-6y71c?fontsize=14&hidenavigation=1&theme=dark&file=/src/index.js
React State
User state

Tuesday - Beginning State

Search:

- Build a parent component (ShowCardList) for ShowCard
    - shows.map((show) => {
        <ShowCard props={show}>
- Build a Search page component
    - <Search>
      <ShowCardList>
    - useEffect for API call? For now pass it a static JSON?

Shows

- Build a parent component (PostList) for PostCard
    - posts.map((post) => {
        <PostCard props={comments}>
- Build a Show page component
    - <ShowBanner>
        <ShowDetails>
      <PostList props={posts}>
- Pass a hardcoded show_id for now. Add a show to the database with comments and stuff?

Wednesday

- Dynamic Router
- Render posts & comments on front-end from back-end
- API calls to backend
- onClick handler for Search
- MyShows page -> static component, pass props

Cleanup

- useVisualMode on MyProfile components (onClick of the Edit button)
- 404 page
- Remove or handle html tags in show descriptions

Pitch

- Introductions: tell them your name & your background! (see below for tips)
    - (45-60 sec total for the whole group, so 15-20 sec/person)
- Inspiration for the project (the User Story/pitch)
    - (30-45 sec)
- Walk-through of the main features including tech stack and challenges
    - (120-150 sec)
- How you worked as a team
    - (30 sec)
- Further developments (in the future, we'd like to add this feature, etc.)
    - (30-60 sec)

A: THE APP

INTRO
You're sitting on the couch, phone in hand, and the most amazing twist happens in your favourite show! What are you going to do? Either you can go to five different websites… Or you open up We.TV to see what the community is saying!

WALK-THROUGH

Homepage
Login
My Shows
Search -> Game Of Thrones
Game Of Thrones show page
Add to Favourites
My shows (Look now I can navigate to the show from here)
Add post
Add post with image "Hi I'm new here" OR "Me after watching Red Wedding"
Like a dummy post
Dislike a dummy post
Comment on a dummy post -> Post is asking "which episode was it where X happened?"
"Oh, I know! Let's reply to that."
Visit Chat
Send a message
Send a mean message from another browser window
"I don't like this community anymore"
Unfavourite the page
My Shows (page is gone)
Search for a chat recommended show
Logout

B: THE DEVS

CHALLENGE: INTUITIVE UI/UX

We wanted a very user-friendly application. UI/UX were very important in the development of the app. Behind the scenes, a lot has to happen for it to be this easy to use.

We had to learn Material.UI
We had to manage state in order to perpetuate data

STACK
Front: ReactJS, Axios, Material-UI
Back: Express + Node JS, Socket.io
Database: PostgreSQL
API: TVMaze, Cloudinary

TVMaze
The main feature of our app is being able to dynamically generate a show page….. Explain the whole thing
Horizontally deployable (again!)

Cloudinary

React / MaterialUI
State / Vertical integration of data

HOW WE WORKED
We worked vertically so that we all had a holistic understanding of the project
Lots of Pair Programming
We all did full-stack implementation, from SQL database requests to CSS

FURTHER DEVELOPMENTS / STRETCH FEATURES

- Spoilers button
- Friends, individual chat, all that social media stuff
- Recommendations based on your favourites
- View parties

ENDING

Thanks for viewing our app presentation. If you have any questions please join us in the session!