**Assignment - Rock the Vote**

**Timeline**

***This is a big project, but we will break it down and tackle bite-size portions each week.***

[Part 1: Week 3 - Initial Backend Setup](#_94ssfajwwr93)

[Part 2: Week 4 - Server Continuation and Frontend setup](#_19dmjntafuvm)

[Part 3: Week 5 - Front-end authentication, and adding security features](#_5yrtyrq7z53f)

[Part 4: Week 6 - User Experience (UX) Improvements](#_jrcvtkeuwge0)

[Part 5: Week 7 - Finishing Touches](#_39iqa5da419s)

**Requirements**

* A login/signup page that is required to visit any part of the application.
* Web page that shows a list of political issues.
  + Each item will have a title
  + Each item will have a description
  + Each item can be upvoted or downvoted.
    - Users should only be able to upvote/downvote once per issue.
  + Items will be ordered by upvotes (the most being at the top).
  + Each item should show the total number of votes
* Include a page that will allow logged in users to add new posts.
* Include a page that will allow logged in users to view all posts they personally made.
* Users can comment on posts (unlimited number of comments per post)
* You will create a back end with the needed 'models' and 'routes' so that all topics, comments, and votes are persistent.

**Suggestions**

1. Context

* Due to the amount of functionality, consider using 2 context providers. One that manages the authentication/user functionality of the app, and another that manages the issues and comments.

1. State

* Due to the amount of state and state update operations you may need, consider using useReducer to manage context state. Using useState will work as well.

1. Relationships

* You will need to manage *one to many* & *many to many* database relationships.
* The *one to many* relationships will be relating the user to issues, and relating the comments to users and to issues.
* The *many to many* relationships will be part of the upvote/downvote process as issues can be upvoted/downvoted by many users, and many users can upvote/downvote many issues. This will be needed to make sure that a "issue" can only be upvoted/downvoted once per user.

## **~~Part 1: Week 3 - Initial Backend Setup~~**

~~Setup a server~~

* ~~Using Mongoose, connect to a mongoDB database.~~

~~Create the following Mongoose Models:~~

* ~~User~~
* ~~Issue~~
* ~~Comment~~

~~Set up the Authentication route - ex. authRouter~~

~~Create CRUD operations:~~

* ~~CREATE~~
* ~~READ~~
  + ~~Find One~~
  + ~~Find All~~
* ~~UPDATE~~
* ~~DELETE~~

## **Part 2: Week 4 - Server Continuation and Frontend setup**

~~Backend:~~

* ~~Install and implement:~~
  + ~~Dotenv~~
    - ~~Remember to prevent the .env file being pushed to GitHub~~
  + ~~Express JSON Web Tokens (JWTs)~~
* ~~Relationships~~
  + ~~A~~ *~~one to many~~* ~~relationship will be relating the user to issues, and relating the comments to users and to issues.~~
  + ~~A~~ *~~many to many~~* ~~relationship will be part of the upvote/downvote process as issues can be upvoted/downvoted by many users~~

Frontend:

* Create a React app
* Build the following components:
  + Login & Signup - ex. Use an AuthForm
  + Profile
  + Issues
* Connect your Frontend & Backend using a proxy
* Create Context for the User - ex. Create a UserProvider function

## **Part 3: Week 5 - Front-end authentication, and adding security features**

*Note: This week’s project requirements are related to the material of Week 5.*

* Saving the User and Token
* Add logging out functionality - reset state & localStorage
* Create Axios HTTP Requests
  + Use Axios interceptor to add User token to request headers
* Encrypt passwords using bcrypt
* Use schema methods to check & remove passwords

## **Part 4: Week 6 - User Experience (UX) Improvements**

*Note: This week’s project requirements are related to the material of Week 5.*

* Frontend errors - If you haven’t yet, implement providing error messages to the User, ex. Incorrect username or password, etc.
* Conditional Views - create at least one conditional view - ex. allow only logged in Users to see a logout option.
* Protected routes - prevent access to URL endpoints, even if manually type in url/endpoint, ex. localhost:3000/profile

## **Part 5: Week 7 - Finishing Touches**

* This week please review your Rock the Vote application and the REQUIREMENTS (above).
* Take the time to finish up any parts of the project that need completed.
* Add in any additional functionality or design you like.

Johnny User:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3N1ZXMiOltdLCJfaWQiOiI1ZWI0OGVkZWZlYWU5MzJlMjAwNDFkMWIiLCJuYW1lIjoiSm9obiBTbWl0aCIsInVzZXJuYW1lIjoiSm9obm55IiwicGFzc3dvcmQiOiJhd2xraWdpbmZhcyIsImJpcnRoZGF0ZSI6MTk2NTExMDQsIl9fdiI6MCwiaWF0IjoxNTg4ODkxMzU4fQ.bmgzeHnnj1dbcxuf8bestpP-K4xd7psh0oUsoIFYxIM",

    "user": {

        "issues": [],

        "\_id": "5eb48edefeae932e20041d1b",

        "name": "John Smith",

        "username": "Johnny",

        "password": "awlkiginfas",

        "birthdate": 19651104,

        "\_\_v": 0

    }

}

Lissa User:

{

    "token": " eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3N1ZXMiOltdLCJfaWQiOiI1ZWI0OGYxMWZlYWU5MzJlMjAwNDFkMWMiLCJuYW1lIjoiTGlzc2FuZHJhIEhlcmlsbGl1bW1lIiwidXNlcm5hbWUiOiJMaXNzYUhybG0iLCJwYXNzd29yZCI6ImFzZGFza2ZhO2wiLCJiaXJ0aGRhdGUiOjIwMDAxMTA3LCJfX3YiOjAsImlhdCI6MTU4ODg5MjExMn0.91AwyUgVvu6-L0XbDbGJhFvHCY0ISdjD3tguZhIipLQ

",

    "user": {

        "issues": [],

        "\_id": "5eb48f11feae932e20041d1c",

        "name": "Lissandra Herilliumme",

        "username": "LissaHrlm",

        "password": "asdaskfa;l",

        "birthdate": 20001107,

        "\_\_v": 0

    }

}

July Summers User:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3N1ZXMiOltdLCJfaWQiOiI1ZWI0OGY0YWZlYWU5MzJlMjAwNDFkMWQiLCJuYW1lIjoiSnVseSBTdW1tZXJzIiwidXNlcm5hbWUiOiJKU3VtbWVycyIsInBhc3N3b3JkIjoic2dzZGdzaHNoc2giLCJiaXJ0aGRhdGUiOjIwMDExMTA3LCJfX3YiOjAsImlhdCI6MTU4ODg5MTQ2Nn0.rxp\_EaimG\_ShX3vbohZBIb0q3-ruzd3k-4YpIFHutLY",

    "user": {

        "issues": [],

        "\_id": "5eb48f4afeae932e20041d1d",

        "name": "July Summers",

        "username": "JSummers",

        "password": "sgsdgshshsh",

        "birthdate": 20011107,

        "\_\_v": 0

    }

}

Jason Mathews:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3N1ZXMiOltdLCJfaWQiOiI1ZWI0OGY4NWZlYWU5MzJlMjAwNDFkMWUiLCJuYW1lIjoiSmFzb24gTWF0aGV3cyIsInVzZXJuYW1lIjoiSk1hdGhldyIsInBhc3N3b3JkIjoic2dzZHdldHdldHdoIiwiYmlydGhkYXRlIjoyMDAzMTIwNywiX192IjowLCJpYXQiOjE1ODg4OTE1MjV9.7Q9FS9llybZFzA9uzDYPq3NKsYJNh-mVLvDRZPXpbEU",

    "user": {

        "issues": [],

        "\_id": "5eb48f85feae932e20041d1e",

        "name": "Jason Mathews",

        "username": "JMathew",

        "password": "sgsdwetwetwh",

        "birthdate": 20031207,

        "\_\_v": 0

    }

}