Backend:

POST “/users/register”

This route validates the form and if validation passes it saves the user to the database if the user doesn’t already exist.

POST “/users/logout”

This route destroys the session, calls the passport-provided logout method and clears the session cookie.

GET “/users/verify”

This route checks to see if the session is still active and returns the corressponding information.

POST “/users/login”

This route is handled by the passport local strategy middleware we define in /backend/config/passport.js

Frontend:

Logout function

This function is triggered from the logout component and defined in the top component app.js, this makes a post request to “/users/logout” This effects the “global” loggedin app.js state variable.

Login function

This function is triggered from the login component and defined in the top component app.js, this makes a post request to “/users/login” This effects the “global” loggedin app.js state variable.

Register function

this function is not relevant to the rest of the website and therefore is defined within the register component. There is an axios post request to “/users/register”

Express-Session: <https://github.com/expressjs/session>

Login function

sessions are stored server side unlike tokens which must be stored client side. we need to connect a session store to avoid memory leaks and persist our user. the most popular are: connect-mongo and connect-redis. connect-redis offers more features that you can explore when you begin configuring heroku like caches or load balancing options. connect-mongo however allows us to use our existing mongo connection, so its convenint. though, both are setup in the same manner.

Session

This attaches the session object to the request object lcoated in route handlers. we configure both our cookie and session through this. “req.session” this is also where we attach our session store.

Passport and passport-local: <http://www.passportjs.org/docs/authenticate/>

Deserialize/ Serialize methods

Session

passport attaches its session on top of express-session. passport strategies allow you to use 3rd party login services or setup up a local authentication strategy.

Login function

these methods must be implemented and attach: user, isAuthenticated(), and logout() to the request object located in route handlers. these methods require a unique key which we can use our mongodb unique id attribute for.

Session