* Full Stack Cat App IV

** Streamline UI and Clean Up Routes

* There is no longer a need for 'views/layouts/login/username.js', when the user logs in, it makes more sense to go to 'views/layouts/third-page.ejs' , the profile view, or if it is a new user, to the what is your age view. Eventually, the app will open on a site page and the login will be a button, and the profile will also be a button that is visible after the user is logged on to the site.
* In routes/auth/login.js in the router.get'/dashboard' route, change res.render to res.redirect(`/landing/${id}`); You will no longer need the name variable, but you will still need the id. You can either use backticks and ES6 template literals or can concat the route:

````

router.get('/dashboard', ensureAuthenticated, async (req, res) => {
    userId = req.user.id
    await User.findById({
        _id: userId
    }, (err, data) => {
        if (err) {
            console.log(err);
        } else {
            let id = data._id;
            res.redirect(`/landing/${id}`);
        }
    });

});


````

* When this is working, in views/layouts/login delete username.ejs, update-profile.ejs and new-profile.ejs .
* In routes/auth delete update.js .
* In routes move update.js into routes/auth .
* In routes/index.js delete const updateRoutes = require('./update') and router.use('/update', updateRoutes) -- otherwise the app will break looking for them.
* In routes/auth/index.js change router.use('/user', updateRoutes) to router.use('/update', updateRoutes)
* All other paths SHOULD stay the same, however test them out.
* When everything is working, save to GitHub.

*** Clean up routes/index.js

* In routes make a folder called profile and in the folder make a file called index.js and another called profile.js
* In routes/profile/index.js const router = require express as Router(), require const profileRoutes from profile.js and when in the profile folder have router use profileRoutes at '/' ('/profile'). Finally, export the router. It should look like this:

````
const router = require('express').Router();
const profileRoutes = require('./profile');

router.use('/', profileRoutes);

module.exports = router;

````

* In routes/profile/profile.js besides bringing in the express Router, bring in both User and CatFancier and ensure Authenticated. The paths for the models and ensureAuthenticated will have to change (add ../ if copying from router/index.js).
* Cut and paste the routes from routes/index.js into routes/profile/profile.js.
* Export the router. routes/profile/profile.js should look like this:

````
const router = require('express').Router();
const CatFancier = require('../../models/CatFancier');
const User = require('../../models/User');
const {
    ensureAuthenticated
} = require('../../passport/auth');

router.get('/landing/:id', ensureAuthenticated, (req, res) => {
    User.findById({
        _id: req.params.id
    }, (err, data) => {
        if (err) {
            console.log(err);
        } else {
            let name = data.name;
            let id = data._id;
            let email = data.email;
            CatFancier.findOne({
````

```
                user_id: id
            }, (err, data) => {
                if (err) {
                    console.log(err)
                } else if (!data || data === undefined) {
                    res.render('layouts/form', {
                        name,
                        id
                    });
                } else {
                    let name = data.name;
                    let age = data.age;
                    let fci = data.favoriteCatImg;
                    let id = data._id;
                    let user_id = data.user_id;
                    res.render('layouts/third-page', {
                        email,
                        name,
                        age,
                        fci,
                        id,
                        user_id
                    });
                }
            });
        }
    });
});

router.post('/catimage/:id', ensureAuthenticated, (req, res) => {
    let id = req.params.id;
    let age = req.body.age;
    User.findById({
        _id: id
    }, (err, data) => {
        if (err) {
            console.log(err);
        } else {
            let name = data.name;
            let user_id = data._id;
            let catFancier = new CatFancier({
                name,
                age,
                user_id
            });
            catFancier.save((err, data) => {
                if (err) {
                    console.log(err);
                } else {
                    let id = data._id;
                    CatFancier.findById({
                        _id: id
                    }, (err, data) => {
                        if (err) {
                            console.log(err);
                        } else {
                            let name = data.name;
                            let age = data.age;
                            let id = data._id;
                            res.render('layouts/other-page', {
                                name,
                                age,
                                id
                            });
                        }
                    });
                }
            });
        }
    });
});

router.post('/third-page/:id', ensureAuthenticated, (req, res) => {
    let src = req.body.src;
    let id = req.params.id;
    CatFancier.findByIdAndUpdate({
        _id: id
    }, {
        $set: {
            favoriteCatImg: src
        }
    }, {
        new: true
    }, (err, data) => {
        if (err) {
            console.log(err);
        } else {
            let id = data.id;
            CatFancier.findById({
                _id: id
            }, (err, data) => {
                if (err) {
                    console.log(err);
                } else {
                    let name = data.name;
                    let age = data.age;
                    let fci = data.favoriteCatImg;
                    let user_id = data.user_id;
                    let id = data._id;
                    User.findOne({
                        _id: user_id
                    }, (err, data) => {
                        if (err) {
                            console.log(err);
                        } else {
                            let email = data.email
                            res.render('layouts/third-page', {
                                name,
                                email,
                                age,
                                fci,
                                id,
                                user_id
                            });
                        }
                    });
                }
            });
        }
    });

});

router.get('/delete/:id', ensureAuthenticated, (req, res) => {
    let id = req.params.id;
    CatFancier.findById({
        _id: id
    }, (err, data) => {
        if (err) {
```

````
            console.log(err);
        } else {
            let user_id = data.user_id;
            User.findByIdAndRemove({
                _id: user_id
            }, (err, data) => {
                if (err) {
                    console.log(err);
                } else {
                    CatFancier.findByIdAndRemove({
                        _id: id
                    }, (err, data) => {
                        if (err) {
                            console.log(err);
                        } else {
                            let name = data.name;
                            res.render('layouts/delete.ejs', {
                                name
                            });
                        }
                    });
                }
            });
        }
    });
});


module.exports = router;
````

* In contrast, routes/index.js should now look like this:

````
const router = require('express').Router();
const siteRoutes = require('./site');
const profileRoutes = require('./profile');
const authRoutes = require('./auth');

router.use('/restOfSite', siteRoutes);
router.use('/profile', profileRoutes);
router.use('/', authRoutes);
````

````
module.exports = router;
````

* routes/index.js should now have the models and ensureAuthenticated removed, and the profileRoutes should be required and used at '/profile'.

Change paths

* All the paths that point to these routes must now be changed. The router/auth/login.js router.get'/landing/:id' route should now redirect to `/profile/landing/${id}` .
* Similarly, the paths in the views must change as well, in views/layouts/updates/partials/cat-img-form.ejs the form actions must now point to "/profile/third-page/<%= id %> . In layouts/form.ejs the action now points to "/profile/catimage/<%= id %>" . In layouts/third-page.ejs the delete button now points to '/profile/delete/<%= id %>' . And in views/partials the other cat-img-form.ejs also needs to be changed.
* Carefully go through the rest of the views and then the routes and add '/profile/' to any route that was formally in routes/index.js (Not in routes/profile/profile.js! Those routes should all start with '/' !).
* For some reason, the image paths in views/partials/cat-img-form.ejs might need to be changed back to <img src="/assets/images/disgusting.gif" /> , that is to say, remove the extra ../
* When everything is working, save to gitHub.

Make Sure That All Routes Are Authenticated

* Make sure that all the routes in routes/profile/profile.js include the ensureAuthenticated middleware.
* In routes/site/index.js include the ensureAuthenticated middleware.
* In the router.post'/' route include ensureAuthenticated between the route and the req, res callback:

````
const router = require('express').Router();
const CatFancier = require('../../models/CatFancier');
const {
   ensureAuthenticated
} = require('../../passport/auth');

router.post('/', ensureAuthenticated, (req, res) => {
````

* Now, unless the user is authenticated he or she cannot get into the site.
* When everything is working, save to GitHub.

Clean up Views

* In views/layouts make a folder called profile.
* Put delete.ejs, form.ejs, other-page.ejs, and third-page.ejs in views/layouts/profile. DO NOT put page.ejs in views/layouts/profile.
* In views/layouts/profile make a folder called partials. Move cat-img-form.ejs from views/partials to views/layouts/profile/partials.
* Move the view/partials folder that now contains messages.ejs and passport-messages.ejs into views/layouts/login.
* views should now only have one folder in it - layouts, which contains four folders: login, profile, site and updates. It should also contain page.ejs
* Now all the paths need to be painstakingly reworked. Start by going through all the routes, particularly the update and profile routes and then go through all the views, to make sure that the partials work. The login/login and registration views will have to be gone over as well, as they contain partials. Here is the new delete render and the new partial for login, for example:

````
//In the layouts/profile folder
res.render('layouts/profile/delete.ejs', { name });
//partials folder is much closer now
<%- include('./partials/passport-message') %>

````

* It might take a while for things to update, especially for all the images to render. Keep working it with existing users and by adding new users. Try out all the routes.
* When everything is working, save to GitHub.

Where next?

* When we navigate to the site (localhost:3000) we should see a portal view with a login button.
* Once we are logged in, we should have access to the site, which should have a profile button.
* There should be a site that uses the variables, user age and favorite cat image, in some sort of activity, probably a game.