

* Full Stack Cat App II

** Reworking the Models

* Both models, User and CatFancier share a common field: name.

* In this part of the exercise the objective is to pass the username from User to CatFancier as well as passing the mongodb document _id associated with that name to CatFancier as user_id.

** Adjusting the Model and Passing a Value From One Model to Another

* In the CatFancier model add a field: user_id set to a string and required. CatFancier should now look like this:

....

```
const mongoose = require('mongoose');
```

```
const CatFancierSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    trim: true
  },
  user_id: {
    type: String,
    required: true,
    trim: true
  },
  age: {
    type: Number,
    required: true,
    trim: true
  },
  favoriteCatImg: {
    type: String,
    trim: true
  }
});
```

```
module.exports = mongoose.model('CatFancier', CatFancierSchema);
```

....

- * Next, we must go into the routes to pass the name and the `_id` from User to CatFancier .
- * In `routes/index.js` change the `router.get '/landing'` route to a `router.get '/landing/:id'` route in order to capture the User model.
- * In `username.ejs` change the `onClick` in 'on to landing page' from `window.location.href='/landing'` to `window.location.href='/landing/<%= id %>'` so that it points to the `/landing/:id` route .
- * In `routes/index.js` use `req.params.id` to let User `findById()` and then get the name and the `mongodb _id` and pass them in the render to `layouts/form` .
- * In `views/layouts/form` remove the name input, and change the form to only ask for the user's age, change the post route to include id so that it is now `"/catimage/:id"` :

....

```
<div class="form-container">
  <h1><%= name %>, What is your age?</h1>
  <form action="/catimage/<%= id %>" method="POST">
    <h3>Please enter your age</h3>
    <input id="age" type="number" name="age" placeholder="Your Age" required/>
    <input type="submit" value="Submit">
  </form>
</div>
```

....

- * In `router.post '/catimage'` change the post route to be `"/catimage/:id"`, then let `id = req.params.id` and `age = req.body.age`, for the moment delete the `name = req.body.name`.
- * With the `id`, `User.findById()` and retrieve the name and the `_id` as `name` and `user_id` .
- * let `catFancier = new CatFancier(name, age, user_id);`
- * `catfancier save()` and with the `data._id` `CatFancier.findById()` and use the data to render 'layouts/other-page', passing in `name`, `age` and `id` (which in this case will be the `CatFancier _id` and not the `User _id`) .
- * Add `ensureAuthenticated` to the route and to all the other routes on the page.
- * The new route should look like this:

....

```
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
            });
          }
        });
      }
    });
  }
});
}
});
});

```

....

* In views/partials (NOT views/layouts/partials) the cat images might need '../' added to the path so that the browser can find them. <button type="submit"></button>

* Next we must fix the delete route so that it deletes from both the User and the CatFancier models.

* Then we must validate the post route so that it does not keep saving the user's data if it's already in the database.

* But first, make sure that everything is working and save it to gitHub.

** Adjusting the Delete route so that it deletes the User from Both Collections

* In routes/index.js at router.get 'delete/:id' use req.params.id for CatFancier to findById() .

* From the data extract the user_id to use for User.findByIdAndRemove() .

* Then in the callback, use the id (req.params.id that should already be defined) for CatFancier.findByIdAndRemove() .

* In the callback for CatFancier.findByIdAndRemove() extract the name and render 'layouts/delete.ejs' while passing it the name.

* The delete/:id route should now look like this:

....

```
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 });
            }
          });
        }
      });
    }
  });
});
```

....

- * Check the database to see if both the User and the CatFancier documents have been removed.
- * Because we no longer need that functionality, let's remove the delete route from the login.
- * In views/layouts/login/username.ejs remove the delete button and two of the `
` .
- * In routes/auth/update.js remove the `removeUser/:id` route .
- * If everything is running, save to gitHub.

** Validate CatFancier Route (Only Save to Route Once)

- * In routes/index.js `router.get('/landing/:id'` in the `User.findById()` if there are no errors, have `CatFancier.findOne()` where the `{ user_id : id }` .
- * If CatFancier cannot find data or the data is undefined, render 'layouts/form' and pass in the name and id from the User.
- * If CatFancier does find data, then pass that data while rendering 'layouts/third-page' .
- * The route should now look like this:

....

```
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;
      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;
          res.render('layouts/third-page', {
```

```

        name,
        age,
        fci,
        id
      });
    }
  });
}
});
});
....

```

* Lastly, there is still a problem with the name update, if a name updates in CatFancier, how to update it in User?

* First, however, if everything is working, save to gitHub.

Updating the 'name' Field in Both User and CatFancier

routes/update.js

* require User

* In router.post('/name/:id' in the CatFancier.findById(), in the else statement, and after the variables, add let user_id = data.user_id .

* After the variables have User.findByIdAndUpdate({ _id : user_id }, { \$set: { name : name } }, { new : true }, (err, data) => {} .

*In the data function, render 'layouts/third-page.ejs' passing in name, id, age, fci .

* The route should now look like this:

....

```

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

```

* If everything is working, copy to gitHub.

What's next?

- * Next, the email field must come from User and be able to be updated in third-page.ejs.
- * Unnecessary routes and views should be cleaned up from the login, so the user only has to update their information once.
- * What is your age, and which is your favorite cat gif should be joined.
- * The routes should be reworked to get some routes out of index.js and move update.js to the auth folder.