CWEB280 LO1,2,3 – Practice Project

Pet Registry

# Setup

1. Add the router to app.js with the path '/pets' [1 mark]

# Pet List

1. The Bootstrap table MUST have the following column headings: [3 marks]  
   Pet Name - links to ?sortby=petname, Pet Type - links to ?sortby=pettype, License - links to ?sortby=license, Owner Name - links to ?sortby=ownername
2. Clicking in the headings will reload the page AND send the 'sortby' query string parameter in the url [2 marks]
3. Display 1 pet per table row. Each pet object will have the following attributes: [3 marks]  
   petname, pettype, license, ownername
4. The pet's license must be link to the pets view page - you MUST also send the 'license' query string parameter in the link [2 marks]

# Pets View

1. if the 'license' is passed in the query string, Display all information about the pet: license, ownername, petname, pettype, petimage [2 marks]
2. petimage contains the name of a file in the public/images folder - you MUST display the image using an img tag [3 marks]
3. if license is not passed or is not found then display 'Pet Not Found' message [1 mark]
4. Display a list pet names from the session [1 mark]  
   The pet name should be in a link to pet view [to this page] that sends the 'license' query string parameter

# Pets Add

HBS file

1. Add form inputs for license, petname, pettype, petimage, ownername [5 marks]  
   petimage is a file input, but it is optional to upload a file - all other fields are required
2. You will need to show any form error messages using [bootstrap validation styles](https://getbootstrap.com/docs/5.0/forms/validation/#server-side) [2 marks]

Router File

1. Add a working GET handler for pets add [2 marks]
2. Add a working POST handler for pets add [2 marks]
3. In the post handler add [> express-validators](https://github.com/validatorjs/validator.js#validators)for the following:
   1. license - must be 8 digits [1mark]
   2. petname && ownername - must not me empty (trimmed) and max 50 characters [1mark]
   3. pettype - must not me empty (trimmed) and max 25 characters [1mark]
   4. petimage - is optional but if uploaded must be a PNG file (no other image formats) and max 2MB in size [3 marks]
4. if any are fields are invalid display the error messages (delete the file if uploaded [fs.unlinkSync](https://nodejs.org/api/fs.html" \l "fs_fs_unlinksync_path" \t "_blank)) [2 marks]
5. All fields valid:
   1. figure out how to save the uploaded filename into the database [BONUS 2 marks]
   2. add the pet to an array in the session - so you can display recently added pets in pets view [2 marks]
   3. redirect to pets view with the 'licence' query string parameter in the url [2 mark]

# Login – LO2 – LO3

1. Add options to the Local Strategy for the names of the login form inputs [1 mark]
2. In the Login POST handler passport.authenticate add options for: [2 marks]  
   the success redirect to the 'pets-add' page and [1 mark], failure redirects to login with the err query string parameter ([Passport Redirects](https://www.passportjs.org/docs/authenticate/#redirects)) [1 mark]
3. In the GET login handler: if a cookie exists for the owners name send the value to the handlebars template [1 mark]
4. In the POST login handler: If the user activates 'Remember me for next time' then set a cookie that last 30 days with the value of the posted ownername [3 marks]

## Pets Add Login changes

1. In the GET and POST handlers for pets add - check if req.authenticatedUser is exists or redirect to pets login [2 marks]
2. If the authenticatedUser has a role of admin - display the ownername input [2 marks]
3. otherwise - render a hidden input for ownername with the value of authenticated user's owenername property [2 marks]