Matthew Zeman
355 Project 2

1) Grey Paper
    a) Setting up mongodb and using mongoose for schemas
        i) Challenge - I needed a way to have a set schema for each form being uploaded, then I learned about mongoose
        ii)

```
//connect to mongodb with mongoose
mongoose.connect('mongodb://localhost:27017/login-app')
```

    b) Keeping the user logged in with cookies
        i) Challenge - figuring out how to set up browser cookies and match them to log in the user and keep them logged in
        ii)

```
//To login users
app.post('/api/login', async(req, res) => {
    const {username, password} = req.body
    const user = await User.findOne({username}).lean()

    if(!user){
        return res.json({ status: 'error', error: 'Invalid username/password'})
    }

    //means username and password combo is successful
    if(await bcrypt.compare(password, user.password)){
        const userId = user._id
        const num = Math.floor(Math.random() * 999999999)
        await User.updateOne(
            { _id: userId },
            {
                $set: { ID: num}
            }
        )
        res.cookie('ID', num)
        return res.json({ status: 'ok', data: "good"})
    }

    res.json({ status: 'error', error: 'Invalid username/password'})
})
```

c) Set up image storage in mongodb
   i) Challenge - figuring out how image storage worked because we have to somehow save it in a location that's not the database
   ii)

```
//Set up storage
const storage = multer.diskStorage({
    destination:'uploads',
    filename: (req, file, cb) =>{
        cb(null, file.originalname)
    },
})

//Set up file location stuff
const upload = multer({
    storage: multer.diskStorage({
        destination: (req, file, cb)=>{
            cb(null, "./uploads")
        },
        filename: function(req, file, callback) {
            callback(null, file.fieldname + "-" + Date.now() + path.extname(file.originalname))
        }
    })
})
```

d) Have individual pages and links for every item
   i) Challenge - wasn't sure how im able to make individual pages for one static page
   ii)

```
app.get('/item/:itemId', async (req, res) => {
    const name = await getJustUserNameFromCookie(req);

    const itemId = req.params.itemId;
    const item = await Recipe.findOne({ _id: itemId }).lean();
    const steps = item.steps.split('<br>').filter(step => step.trim() !== '');

    res.render('items', {user: name, item: item, steps: steps});
});

app.get('/edit/:itemId', checkUser, checkIfCurrentUser, async (req, res) => {
    const name = await getJustUserNameFromCookie(req);

    const itemId = req.params.itemId;
    const item = await Recipe.findOne({ _id: itemId }).lean();
    const steps = item.steps.split('<br>').filter(step => step.trim() !== '');

    res.render('edit', {user: name, item: item, steps: steps});
});
```

e) Editing and deleting Recipes

i) Challenge - The user not only had to be logged in, but they had to own the recipe they are trying to edit

ii)

```
async function checkIfCurrentUser(req, res, next) {
    try {
        let myCookieValue = req.cookies.ID;
        let user = await User.findOne({ ID: myCookieValue }).lean();
        let recipes = await Recipe.find({ username: user.username }).lea
        const itemId = req.params.itemId;

        let itemFound = false;
        for (let recipe of recipes) {
            if (itemId === recipe._id.toHexString()) {
                itemFound = true;
                break;
            }
        }

        if (itemFound) {
            next();
        } else {
            next();
            res.redirect("/myRecipes")
        }
    } catch (error) {
        console.error(error);
        res.status(500).send("Internal Server Error");
    }
}
```

f) Using ejs rendering every recipe depending on meal type

i) Challenge - Using ejs javascript in the HTML format

ii)

```
<div class="container">
  <div class="row">
    <% recipe.forEach(recipeItem => { %>
      <div class="col-4 box">
        <a href="item/<%= recipeItem._id %>"><img src="<%= recipeItem.image %>">
        <h3><%= recipeItem.name %></h3><br></a>
      </div>
    <% }) %>
  </div>
</div>
```

g) Getting ejs information into javascript
   i) Challenge - This one took me a little bit, I'm not allowed to use ejs format in javascript so getting the information in their to manipulate it took a little figuring out
   ii)

```
<script>
  //get all recipes to javascript
  var recipes = []
  var mealType = []
  var mealName = []

  '<% allRecipes.forEach(function(recipe) { %>'
        recipes.push('<%= recipe.image %>')
        mealType.push('<%= recipe.meal %>')
        mealName.push('<%= recipe.name %>')
  '<% }); %>'
</script>
```

h) Changing images in index page
   i) Challenge - problem here was to have two images on top of each other but also keep the rest of the page looking the same
   ii)

```
<div class="container-image">
    <img class="top-image" id="image1" src="<%= allRecipes[randomIndex].image %>">
    <img class="bottom-image" id="image2" src="<%= allRecipes[randomIndex3].image %>">
</div>
```

i) When uploading and removing steps from the recipe, it has step numbers that are dynamic
   i) Challenge - The problem here was every time you deleted a recipe it wouldn't change the step it was, so I had to go through every step and change it
   ii)

```
function deleteStep(clickedID){
    var div = document.getElementById('list' + clickedID);

    div.remove();

    // Update the numbering of the remaining elements
    var stepLists = document.querySelectorAll('.stepList');

    for (var i = 0; i < stepLists.length; i++) {
        var stepText = stepLists[i].querySelector('.stepText p');
        var deleteButton = stepLists[i].querySelector('.deleteButton button');
        var newIndex = i + 1;

        stepText.textContent = newIndex + ") " + stepText.textContent.split(') ')[1];

        // Update IDs
        stepLists[i].id = 'list' + newIndex;
        stepText.parentNode.id = 'step' + newIndex;
        deleteButton.parentNode.id = 'list' + newIndex;

        // Update onclick attribute
        deleteButton.setAttribute("onclick", "deleteStep('" + newIndex + "')");
    }
}
```

j) When on your profile page, you can search for the recipe by name
   i) Challenge - figuring out how to change the page to only show what you want it to show
   ii)

```
function Search() {
  // Declare variables
  var input, filter, ul, li, div, i;
  input = document.getElementById("mySearch");
  filter = input.value.toUpperCase();
  ul = document.getElementById("myMenu");
  li = ul.getElementsByTagName("li");

  let searchLength = document.getElementById("searchLength");
  let num = 0

  for (i = 0; i < li.length; i++) {
    div = li[i].getElementsByTagName("div")[0];
    if (div.innerHTML.toUpperCase().indexOf(filter) > -1) {
      li[i].style.display = "";
      num++
    } else {
      li[i].style.display = "none";
    }
  }

  searchLength.innerText = num
}
```