CSE 2102: Lab 5

Tom McCarthy

Code located in my repo: https://github.com/tommcc314/CSE-2102-Software-Engineering

Feature Summary:

- Utilized Thymeleaf's features for dynamic content generation, such as iteration, conditional display, and passing objects/attributes.
- Created REST endpoints for the signup page.
- Quiz page now displays quizzes from a question list object.
- Added classes to support these operations with corresponding unit tests.
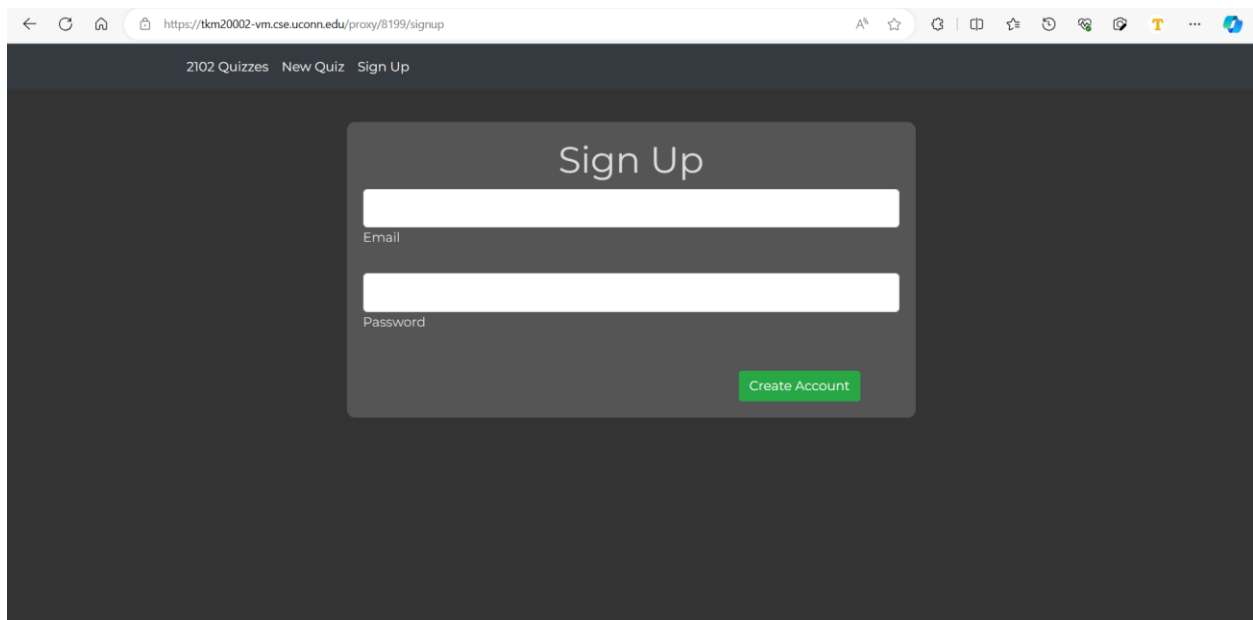
Sign Up Page:

I added a controller class for the sign up page which contains endpoints specific to the page. The endpoints are:
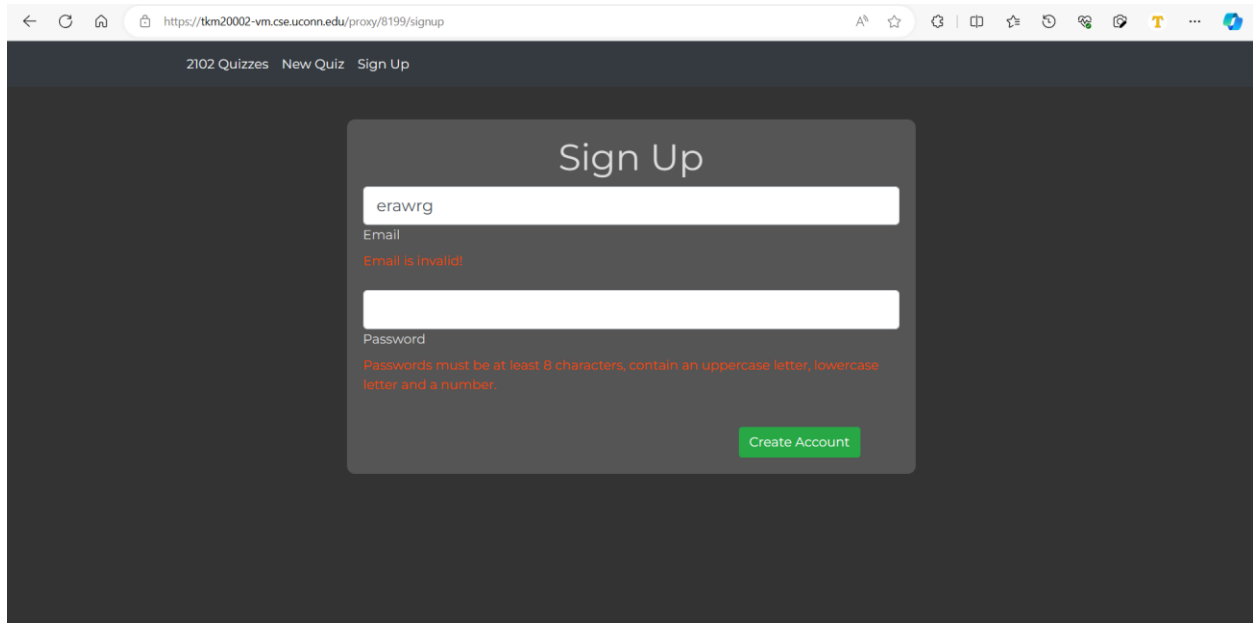
- GET /signup returns the sign up page with an attribute creds which will be passed as an object to the POST endpoint when the email and password are filled out.
- POST /signup takes as input a LoginCredentials object. It creates instances of an EmailValidator and PasswordValidator and calls the corresponding validate functions. It passes the outputs of these as attributes back to the model. In the Sign Up page, there are two text elements for validation errors which will display according to the values of the attributes.

Screenshot of GET /signup:
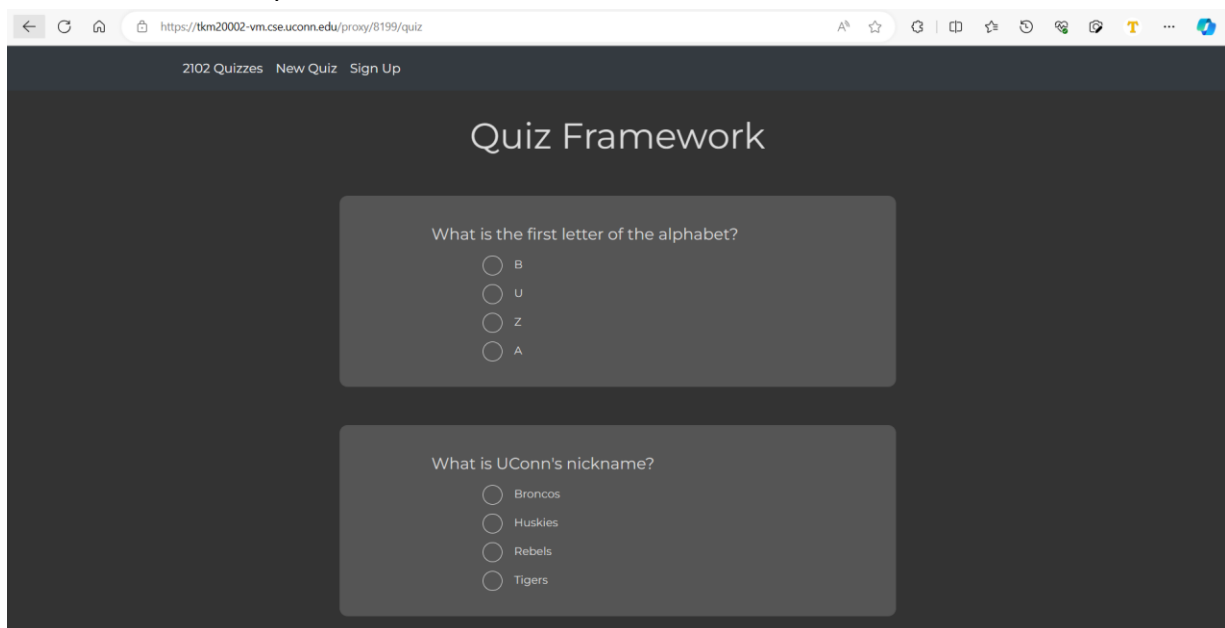
Screenshot of POST /signup with invalid email/password:

Quiz Page:

I added a controller class for the quiz page which contains the quiz endpoint:

- GET /quiz returns the quiz page with questions in the QuestionList. It creates a new QuestionList object and passes it as an attribute to the returned model. The Quiz page contains a thymeleaf iteration where new question elements are generated for each Question in QuestionList.

Screenshot of GET /quiz: