SE 4200: Web Application Development II

Assignment: Midterm Project

Requirements

Create a web application that is of your own creative design. Your application should fulfill the following:

• Design and Ideation

- At least one well-defined resource to be supported throughout your application.
- Documents resulting from the creative and visual design process. At a minimum, this should include:
 - A complete set of low-fidelity wireframes.
 - A complete storyboard representation of your wireframes.
 - A complete set of high-fidelity mock-ups.

• RESTful Web Service using Node.js and Express

- Implement API endpoints for at least 4 distinct RESTful actions between all of your resources.
- Implement data persistence using MongoDB and the Mongoose ODM.
- Implement basic server-side data validation within your Mongoose model(s) and API endpoints.
- Implement user registration: create a simple user model using Mongoose, and use the bcrypt library for Node.js to hash user passwords.
- Implement user authentication: use the express-session middleware to enable session-based authentication.
- Implement basic user authorization: write a custom middleware function that implements basic logic for your own authorization rules. For example, you might simply restrict access to all data unless the user is authenticated, or you might establish record ownership for a resource and restrict access to only those records that are owned by the authenticated user.

• Client Application using Vue.js

- Build an interactive, single-page user interface for all of the endpoints provided by your web service.
- Implement basic client-side data validation for all create and update actions.
- Implement the requests necessary to communicate with your web service for registration, authentication, and retrieving authorized resource data.
- \circ Implement UI panels that allow the user to register and authenticate, as well as the UI elements and logic to display them appropriately.
- Display appropriate messaging for error conditions, including user validation errors and authentication failures.
- Implement the logic necessary to reveal and hide UI elements as part of the authentication and authorization user flows.

• Finishing Touches

- Deploy your finished application using an approved platform, including both the server and client applications.
- Create documentation detailing your resource and its attributes, data model/schema, and all REST endpoints.
 - Include the URL(s) to your deployed application, including both client and server (if deployed separately).

• Use Markdown and save the document as README.md at the base of your Git repository.

Keep the following suggestions and requirements in mind:

- Before you start writing the code for your application, first generate an idea for the application you want to create. Your idea should establish purpose and usefulness, and should be marketable (either for profit or not for profit).
- Translate your idea into a visual representation of the application by creating wireframes, storyboards, and mockups, either on paper or using software.
- Your application should be styled (using appropriate color, imagery, layout, typography, etc.) to look professional. You will be graded based on the effort you apply to create an application that is presentable to a client or customer.
- Your web service and client applications should conform to appropriate protocols, standards, and design patterns, including HTTP, REST, CORS, and MVC. This requires the separation of controller, model, and view logic within both your web service and client applications.
- All data communicated between the client application and the web service should be implemented using Ajax requests.
- While working on your project, you may leverage concepts and techniques from online tutorials and other resources, but the idea that you implement should originate from your own creative process. Your idea should differ significantly from any class or online example.
- You may submit an out-of-class project for this assignment, provided that: 1) the project is developed independently by you, within the timeframe of this current semester, and 2) the project is not submitted for credit for any other assignment, in any other course, in any past, present, or future semester. Violating one or more of these conditions constitutes cheating.

Resources

- Express Documentation
- <u>Vue.js Guide</u>
- Vue.is API
- MongoDB Shell Documentation
- Mongoose Documentation
- bcrypt library documentation
- express-session library documentation

Submission

- 1. Submit your project using Git and GitHub. Start by creating a repo for this assignment here.
- 2. Show your completed assignment to the instructor during class or office hours to receive credit.