

Important details about 401 from the document

- Use MVC for serving main content pages
- Implement microservice to serve comments, using database-per-service pattern.
- Use event sourcing to update the comment service.
- Use CQRS or another architecture pattern as appropriate.
- Incorporates both extra requirements (instead of just one)
 - an extra architecture pattern (e.g. CQRS, or pubsub)
 - deployed to a cloud provider (e.g. Amazon or GCP)
- Using hosted source control with per-user change history (e.g. GitLab)
- Application gracefully degrades if the comment service is shut down
- Application has a working search feature
- Application supports nested comments or comment threads

To do

- Use MVC for serving main content pages
- Implement microservice to serve comments, using database-per-service pattern.
- Use event sourcing to update the comment service.
- Use CQRS or another architecture pattern as appropriate.
- Incorporates both extra requirements (instead of just one)
 - an extra architecture pattern (e.g. CQRS, or pubsub)
 - deployed to a cloud provider (e.g. Amazon or GCP)
- Using hosted source control with per-user change history (e.g. GitLab)
- Come up with types of colors and types of fruits
- Need to create an image for each. Need a generic fruit shape, with color filled in.

We'll use GitHub to share our code and IntelliJ as the programming language

Phase one: The goal is to get the Login and Registration page working, as shown:

The image shows two wireframe diagrams for web pages. The left diagram, titled 'Login page', contains a 'Username:' label with a text input field, a 'Password:' label with a text input field, and a blue button labeled 'Register Account'. The right diagram, titled 'Registration page', contains a 'Username:' label with a text input field, a 'Password:' label with a text input field, a 'Re-type password:' label with a text input field, a blue button labeled 'Submit', and another blue button labeled 'BackToLogin'. Below these buttons is a placeholder for a system message, indicated by the text '<System message displayed here>' and an example: 'For example, "Registration failed. Username already in use"'.

Here are the tasks needed to fulfill this requirement:

Write code for an MVC pattern to serve pages:

- 1) Model will be responsible for modifying data (username and password)
- 2) View will be responsible for the GUI components (Display the login and registration page)
- 3) Controller will communicate between View and Model

Note, here is an example of code for an MVC pattern:

<https://www.geeksforgeeks.org/mvc-design-pattern/>

4) Use MySQL to maintain info in the database. Info maintained will be determined by info in the Model.

5) Put it all together and deploy to a cloud provider (e.g. Amazon or GCP)

After completing phase one, the following should work:

- Display the login and registration page
- Users (at least 2) should be able to register and have their info stored on the database
- Users (at least 2) should be able to log in by verifying the info in the database