Due Jan 30 by 11:59pm

(I)

Grades

Zoom

Billing

Attendance

Student Support

Career Services

Points 100

Model-View-Controller (MVC) Challenge: Tech Blog Writing about tech can be just as important as making it. Developers spend plenty of time creating new applications and debugging existing codebases, but most developers also spend at least some of their time reading and writing about technical concepts, recent

advancements, and new technologies. A simple Google search for any concept covered in

Submitting a text entry box

this course returns thousands of think pieces and tutorials from developers of all skill levels! Your challenge this week is to build a CMS-style blog site similar to a Wordpress site, where developers can publish their blog posts and comment on other developers' posts as well. You'll build this site completely from scratch and deploy it to Heroku. Your app will follow the MVC paradigm in its architectural structure, using Handlebars.js as the

templating language, Sequelize as the ORM, and the express-session npm package for authentication.

User Story

AS A developer who writes about tech I WANT a CMS-style blog site SO THAT I can publish articles, blog posts, and my thoughts and opinions

Acceptance Criteria

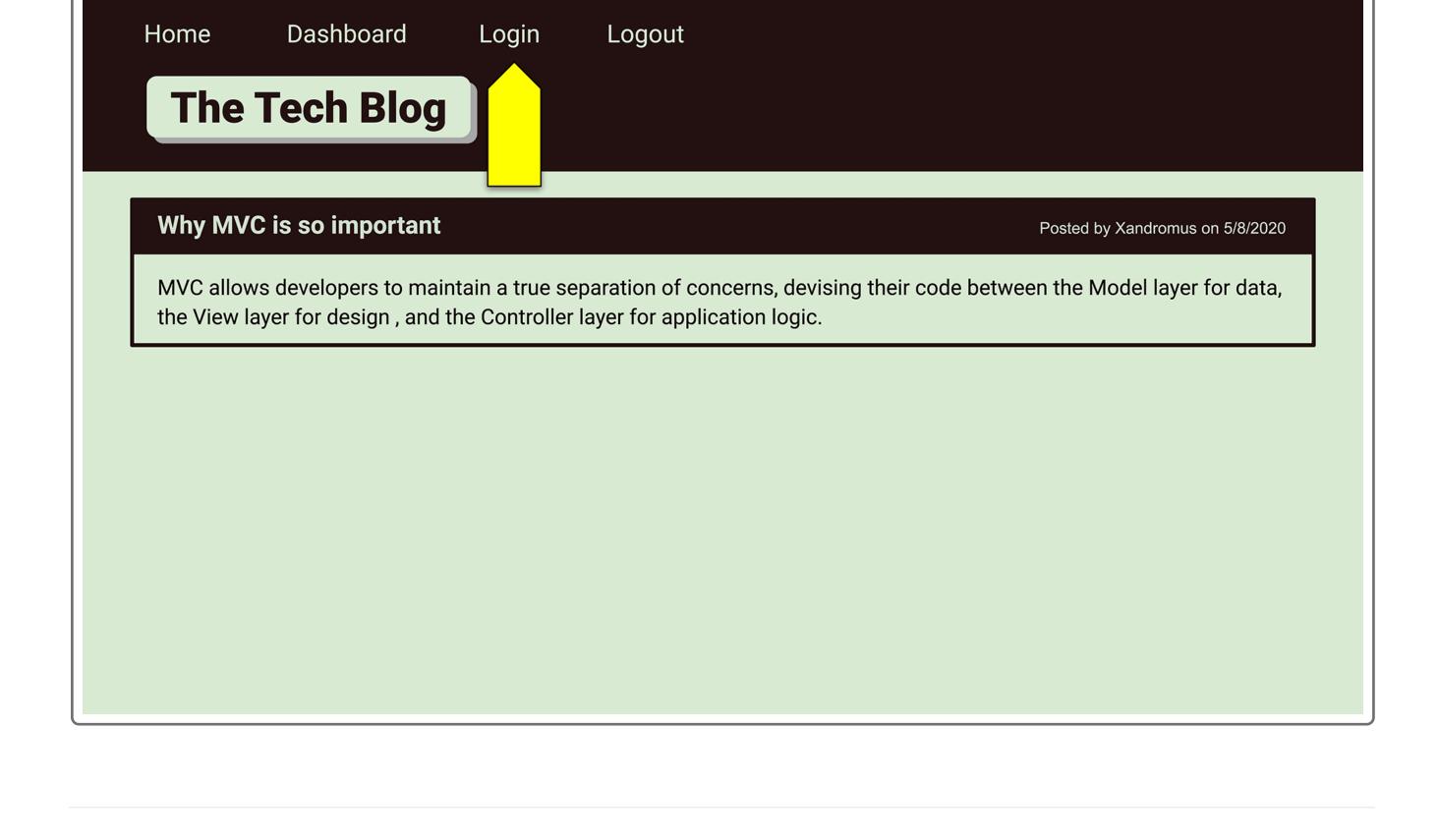
```
GIVEN a CMS-style blog site
WHEN I visit the site for the first time
THEN I am presented with the homepage, which includes existing blog posts if any have been posted; navigation links for
WHEN I click on the homepage option
THEN I am taken to the homepage
WHEN I click on any other links in the navigation
THEN I am prompted to either sign up or sign in
WHEN I choose to sign up
THEN I am prompted to create a username and password
WHEN I click on the sign-up button
THEN my user credentials are saved and I am logged into the site
WHEN I revisit the site at a later time and choose to sign in
THEN I am prompted to enter my username and password
WHEN I am signed in to the site
THEN I see navigation links for the homepage, the dashboard, and the option to log out
WHEN I click on the homepage option in the navigation
THEN I am taken to the homepage and presented with existing blog posts that include the post title and the date created
WHEN I click on an existing blog post
THEN I am presented with the post title, contents, post creator's username, and date created for that post and have the
WHEN I enter a comment and click on the submit button while signed in
THEN the comment is saved and the post is updated to display the comment, the comment creator's username, and the date
WHEN I click on the dashboard option in the navigation
THEN I am taken to the dashboard and presented with any blog posts I have already created and the option to add a new b
WHEN I click on the button to add a new blog post
THEN I am prompted to enter both a title and contents for my blog post
WHEN I click on the button to create a new blog post
THEN the title and contents of my post are saved and I am taken back to an updated dashboard with my new blog post
WHEN I click on one of my existing posts in the dashboard
THEN I am able to delete or update my post and taken back to an updated dashboard
WHEN I click on the logout option in the navigation
THEN I am signed out of the site
```

The following animation demonstrates the application functionality:

Mock-Up

THEN I am able to view comments but I am prompted to log in again before I can add, update, or delete comments

WHEN I am idle on the site for more than a set time



Your application's folder structure must follow the Model-View-Controller paradigm. You'll need to use the express-handlebars package to use Handlebars.js for your Views, use

Getting Started

the MySQL2 and Sequelize packages to connect to a MySQL database for your Models, and create an Express.js API for your Controllers. You'll also need the <u>dotenv package</u> to use environment variables, the <u>bcrypt package</u> \implies to hash passwords, and the <u>express-session</u> \implies and <u>connect-session-sequelize</u> \implies

packages to add authentication. NOTE

The <u>express-session</u> ⇒ package stores the session data on the client in a cookie. When you are idle on the site for more than a set time, the cookie will expire and you

will be required to log in again to start a new session. This is the default behavior and you do not have to do anything to your application other than implement the npm package.

NOTE

Grading Requirements

If a Challenge assignment submission is marked as "0", it is considered incomplete and will not count towards your graduation requirements. Examples of incomplete

submissions include the following: A repository that has no code

- A repository that includes a unique name but nothing else A repository that includes only a README file but nothing else
- A repository that only includes starter code
- This Challenge is graded based on the following criteria:

Technical Acceptance Criteria: 40% • Satisfies all of the preceding acceptance criteria plus the following:

Application's folder structure follows the Model-View-Controller paradigm. Uses the <u>express-handlebars</u> ⇒ package to use Handlebars.js for your Views.

- Application must be deployed to Heroku.
- **Deployment: 32%** Application deployed at live URL.
- Application loads with no errors.

GitHub repository contains application code.

Application Quality: 15%

Application GitHub URL submitted.

- User experience is intuitive and easy to navigate. User interface style is clean and polished.
- Application resembles the mock-up functionality provided in the Challenge instructions.
- **Repository Quality: 13%**
 - Repository has a unique name. Repository follows best practices for file structure and naming conventions.
 - Repository follows best practices for class/id naming conventions, indentation, quality comments, etc.
 - Repository contains a quality README file with description, screenshot, and link to deployed application.

The URL of the functional, deployed application.

NOTE

◆ Previous

Repository contains multiple descriptive commit messages.

How to Submit the Challenge You are required to submit BOTH of the following for review:

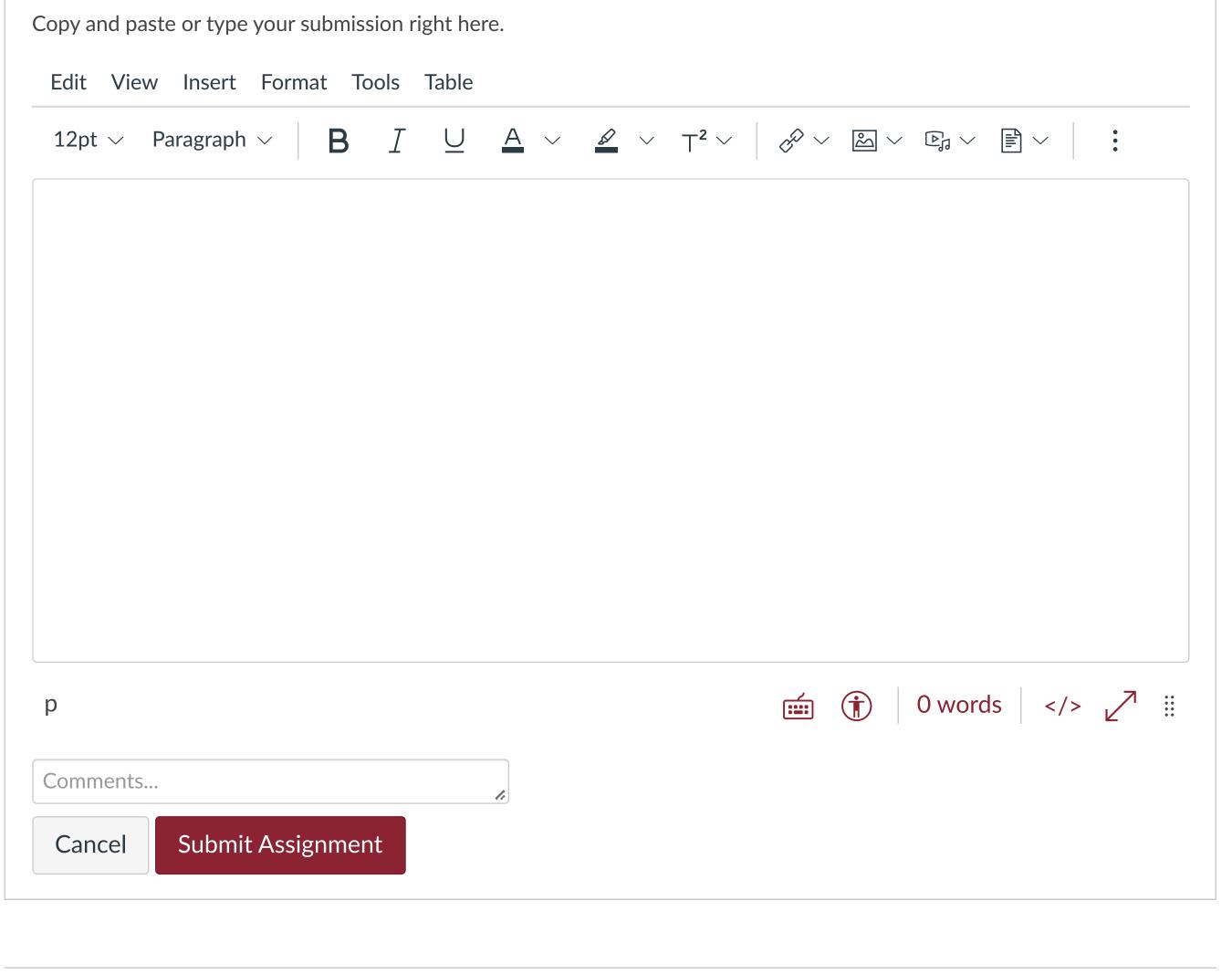
- The URL of the GitHub repository. Give the repository a unique name and include a README describing the project.

certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next Module.

You are allowed to miss up to two Challenge assignments and still earn your

about your feedback, please notify your instructional staff or the Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the Resubmit Assignment button to upload new links. You may resubmit up to three times for a total of four submissions. **Text Entry**

Comments are disabled for graded submissions in BootCamp Spot. If you have questions



© 2023 edX Boot Camps LLC

Next ▶

Submission × Not Submitted! **Submission Details** Grade: 0 (100 pts possible) Graded Anonymously: no No Comments