**Instructions for assignment completion**

1. All hard coded values should be stored in config files.
2. Users who are not authenticated, should not be able to access the application features.
3. E2E test suite along with unit test cases for both the Front End and Back End should be present.
4. Application should be Responsive. User Interface should be optimized for every device from mobile to tablets to desktop.
5. All Back End routes must be protected, unauthenticated user should not be able to access them.
6. You can use any Authentication strategies like Local, OAuth2.0 or JWT

7 . Multilingual support can be implemented. It’s optional requirement not mandatory.

1. Error handling should be done properly for the whole application.
2. README.md should be updated according to your application.
3. Data must be persistent, store it in some database.
4. Dashboard will have 5 topics by default. These topics should be retrieved from database.
5. Confirmation message should be shown where it is necessary for example when user is posting any question on a topic then after creation it should show a confirmation message.

NOTE: Alerts and console.log will not be used in the entire application to show error or confirmation messages.

**As a User, I should be able to register into the system**

By default login page should come on application start up. There will be a register button on login page. On clicking on register button, it should redirect to registration page.

1. User should be able to register into the system
2. Registration page should have firstName, lastName, emailAddress, and password and confirm password fields.
3. All input fields on registration page is mandatory.
4. Error handling should be done for all input fields, for example name can not be a numeric value, etc.
5. On clicking register button, user will be registered and a confirmation message will be shown to the user.

**As a User, I should be able to login into the system**

By default login page should come on application start up.

1. User should be able to login into the system.
2. Login page will have email ID and password fields to login into the system
3. Error handling should be done for all input fields, for example, blank values should not be submitted to the back end.
4. If username and password does not match then show related error message.

**As a User, I should be able to view list of 5 topics in dashboard**

By default when a user logins into the application then a list of 5 topics should be displayed in form of cards

1. Card should have selector .topic-card.
2. It should display the name of the topic in card title and should give selector as .topic-name.
3. It should display the description of the topic in card content and should give selector as .topic-description
4. Card should have a button viewTopicDetails with a selector .view-topic-details.
5. On clicking the button, it should navigate to view topic-detailswhich displays list of questions posted in that particular topic.

**As a User, i should be able to view List of questions posted in a particular topic**

1. Once clicked on a particular topic, it should navigate to next view i.e /topic-details.
2. In topic-details view user should be able to enter question in an input text box with id selector #question-field.
3. User should be able to post new question by clicking a button with id selector #question-button.
4. User should be able to view all posted questions in form of cards.
5. Card should have selector as .question-card.
6. Card should display question asked and should give selector as .question-description.
7. Card should display the date on which the question was posted and should give selector as .question-date.
8. Card should have buttons Delete question with selector .delete-question and view comments with selector .view-comments.

**As a User I should be able to view list of comments posted in a particular question.**

1. Once clicked on button view comment of a particular question, it should navigate to next view i.e /question-details.
2. In question-details view user should be able to post comment in an input text box with id selector #comment-field.
3. User should be able to post new comment by clicking on a button with id selector #comment-button.
4. User should be able to view all posted comments in form of cards.
5. Card should have selector as .comment-card.
6. Card should display comments posted and should have selector as .comment-description.
7. Card should display the date on which the comment was posted and should have selector as .comment-date.
8. Card should have buttons Delete comment with selector .delete-comment.