Querer

Contents

1 Application Features

1.1 Login

- When an user logins for the first time the user need to give an username for account creation.
- The accounts default password is the username of the user.
- For authentication and cookies implemenattion referecnes are :
  - Authentication reference
  - Authentication and Cookie reference video
  - Cookie download and installation

1.2 Auto Complete Search

- Auto complete is implemented using $LIKE$ operator in SQL.
- The given input is checked with the existing data using the $LIKE$ operation whether the input is a substring or not.
- If it is a substring of some data then all the $LIKE$ tuples are shown in the boxes for selection. We can even set a $LIMIT$ on the number of boxes shown.

1.3 Create Posts

- An user need to give Title, Body, Tags of the post as input which is then passed on to trigger written for post insertion for furthur validation.
- Only those tags are taken that are part of a *tags* table.
- A particular post is identified by Post ID.

1.4 Answer to Posts

- This is implemented as comments in the application.
- An user need to give Body for the comment, which is the answer to the post.
- An answer to a particular post is identified by Comment ID.

1.5 Edit Posts

- User can update the Title, Body, Tags of a post and he can mark the question for to stop taking furthur answers.
- User can update the Body of a comment.
- User can delete a post or comment which is posted the user.

1.6 Other

- If a post or comment is upvoted or downvoted then the corresponding user's upvotes and downvotes are changed accordingly.
- If a post or comment is deleted the upvotes and downvotes of the user is setted accordingly.

2 Overall Application Architecture

- Our CQA application uses **MVC** architecture.
- Using this architecture we can clearly separate the business logic, Ul logic and input logic.
- This design pattern separates an application into three main logical components Model, View, and Controller.
  - **Controller** : This is the component that enables the interconnection between the views and the model. It process all the business logic and incoming requests.
  - **View** : The View component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component.
  - **Model** : The Model component corresponds to all the data-related logic that the user works with. Model can add or retrieve data from the database. It responds to the controller's request because the controller can't interact with the database by itself.

3 Tech Stack

- **Frontend** : HTML, CSS, JavaScript, Bootstrap
- **Backend** : JavaScript, ReactJS
- **Backend Frameworks** : NodeJS, ExpressJS
- **Database** : PostgreSQL