Topic: Milestone 8 – Comment and rate

Date: July 17, 2019

Revision: 1

Team: Matt Sievers

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Team Member | Hours Worked | Hours Remaining |
| Update getContent.php  1. Add comment btn to take user to the comment.php new form  2. Comments should now appear below the post they belong to.  3. Add some sort of numerical rating 1-5 (stars maybe)  4. Add viewComments.php  5. Add new btn to view comments, btn value should contain the current comment count for that post.  6. Display avg rate number on post.  7. update pop posts to display the highest rated posts. | Matt Sievers | 4 hrs | 0 |
| update posts table  1. Add FK for postId to postId in comment.  2. update post\_rate to be a float for new avg rate. | Matt Sievers | .5 hrs | 0 |
| Create comment table  1. add new post\_comments\_content to table  2. add new post\_rating and do an average of rates.  3. add comment date/time | Matt Sievers | .5 hours | 0 |
| Utils  1. update getPosts to also now return comments as part of the array.  2. addComment() to update a post with the comment content.  3. getComments() - retrieve all comments.  4. getCommentsCount = retrieve comment count to display on view comments button on getContent.php  5. update updateRate to now update the rate table  6. update getRate to get avg of rates from teh rate table  7. Create getHighestRatedPosts() to get the posts with the highest rating to display them for popular posts. | Matt Sievers | 2 hours | 0 |
| comments.php  1. Create new page to allow user to post similar to the rest (update and create).  2. Title is not needed.  3. Post should be posted to the comments table with a reference to the originalPostId. | Matt Sievers | 2 hours | 0 |
| viewComments.php  1. Filter and sort the comments based on original PostId. Display this similar to admin page to approve posts.  2. Add original post to header for reference  3. Add blog home btn to return to home page. | Matt Sievers | 2 hours | 0 |
| Create rate table  1. Should have originalPostId (FK), date, rate, ID(PK)  2. Use this table to get an avg of the rates by postId and update the rate in posts to be that avg amount. | Matt Sievers | .5 hours | 0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

GIT/Bitbucket URL: <https://bitbucket.org/mSievers82/cst126/commits/all>

Hosting URL: <https://cst126milestone.azurewebsites.net/>

Peer Reviews: NA

Install Instructions: site is now hosted on azure, no installation required locally. Admin account for login: username: admin/password: Password1

General Technical Approach: For this milestone, I wanted to create a “poor mans” rating system. I didn’t want to over complicate it with some addon or template found on the web, I wanted to challenge myself to see if I could complete it within the confines of the native language. I settled on radio buttons and then did some research on how to trigger a submit action on change. For the onchange.form.submit() to work perfectly for this.

Key Technical Design Decisions: I decided to have a button to view the comments as any attempt to add to that form was breaking it. Added a button to view comments and added the number of comments per post to the button value. Also needed to create some new tables to have a rolling count of rates per post to get an average. Finally meant breaking up my two-table design to offer some more data manipulation.

ER Diagram: Located in the repo /milestone/milestone 8 ER Diagram

DDL Scripts:

create schema cst126milestone collate utf8\_general\_ci;

create table user\_info

(

userId int auto\_increment

primary key,

userName char(50) not null,

firstname char(50) not null,

middleName char(50) null,

lastName char(50) not null,

nickName char(50) not null,

email1 char(50) not null,

email2 char(50) null,

address1 char(50) null,

address2 char(50) null,

city char(50) null,

state char(50) null,

zipCode char(5) null,

country char(50) null,

userBanned char default 'n' null,

userDeleted char default 'n' null,

password varchar(255) not null,

confirmPassword varchar(255) not null,

roleName char(50) default 'Spectator' not null,

constraint user\_info\_email1\_uindex

unique (email1),

constraint user\_info\_userName\_uindex

unique (userName)

);

create table posts

(

post\_id int auto\_increment

primary key,

post\_title varchar(255) not null,

post\_content longtext not null,

post\_date datetime default CURRENT\_TIMESTAMP not null,

post\_accepted tinyint(1) default 0 null,

post\_denied varchar(10) null,

post\_author char(50) null,

post\_rate float default 1 not null,

constraint posts\_user\_info\_userName\_fk

foreign key (post\_author) references user\_info (userName)

on update cascade on delete set null

);

create table comments

(

comment\_id int auto\_increment

primary key,

comment\_content longtext not null,

comment\_date datetime default CURRENT\_TIMESTAMP not null,

post\_id int not null,

constraint comments\_posts\_post\_id\_fk

foreign key (post\_id) references posts (post\_id)

);

create table rate

(

id int auto\_increment

primary key,

post\_id int not null,

rate\_date datetime default CURRENT\_TIMESTAMP not null,

rate tinyint(1) not null,

constraint rate\_posts\_post\_id\_fk

foreign key (post\_id) references posts (post\_id)

);

Sitemap Diagram: NA

User Interface Diagrams: NA

Other Documentation: NA