Project summary

Daler Asrorov

ShareToLive

ShareToLive is an app that allows the users to share their thoughts, experiences, and events. They can share their location, write a post, and add a youtube video that they could upload on YouTube or a music video from YouTube. The basic idea is to be able to share your thoughts and experiences instantly, and all the stories have different categories and tags, which can be queried in the application.

Instructions

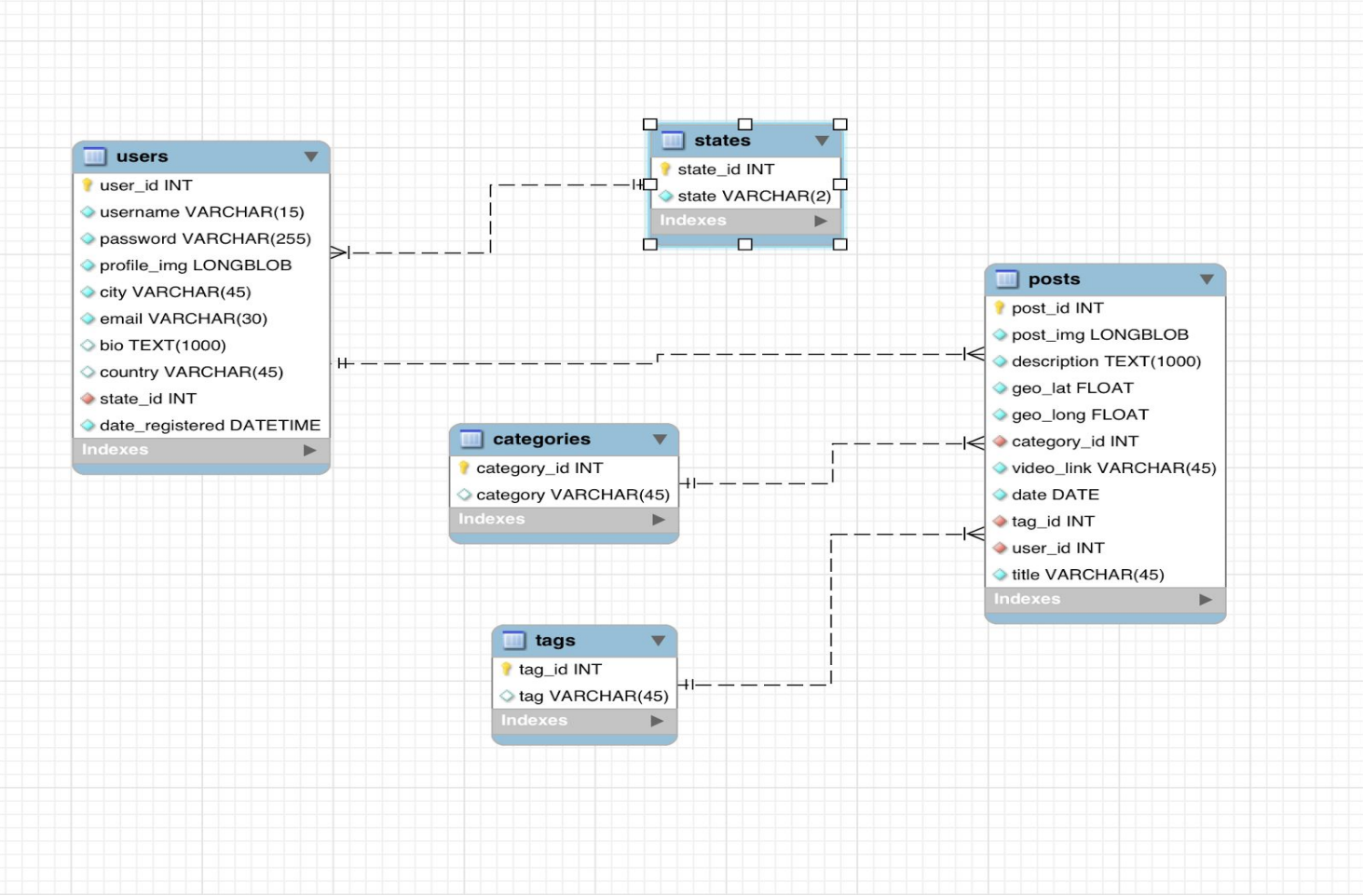
The application has a tutorial, which you can see after you log in. If you click on the '?' icon, you will be directed to the tutorial which explains what each page does and how to use it.

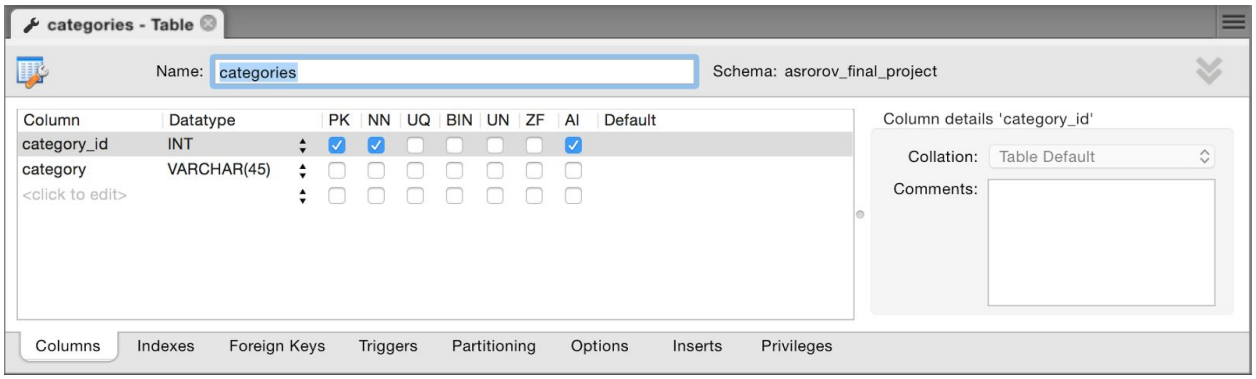
Credentials

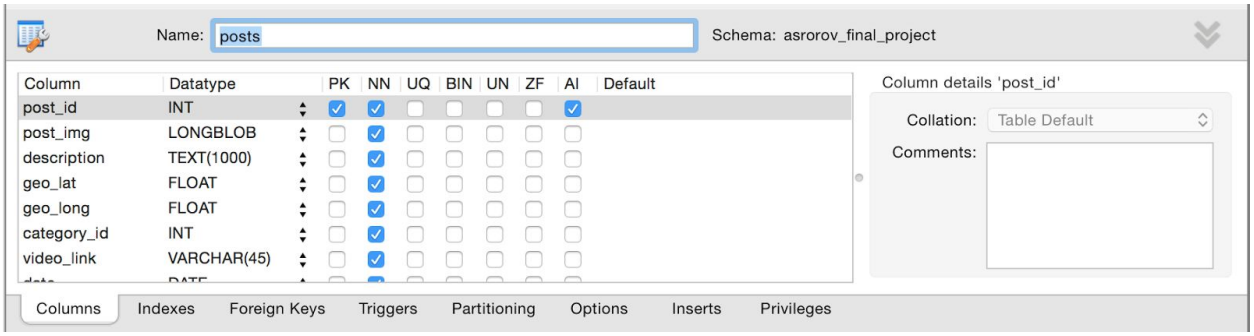
Two options

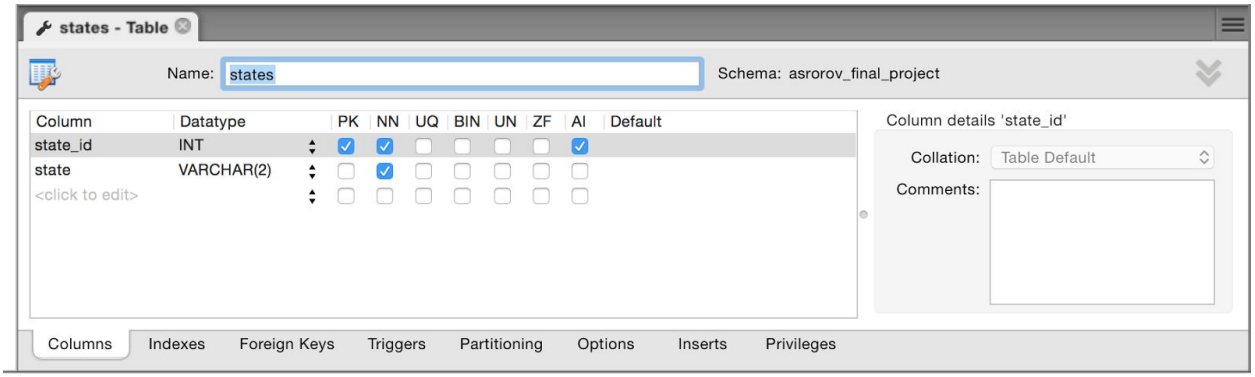
1. You can signup and immediately start checking out the application using your own credentials, or,
2. you can enter the following credentials to be one of the users:
    - username: ttrojan
    - password: usc2015

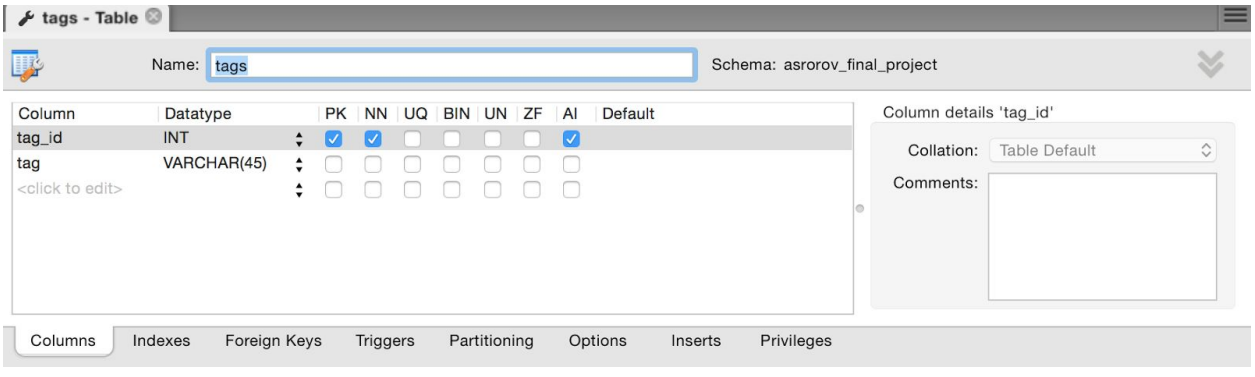
Database Design and Model

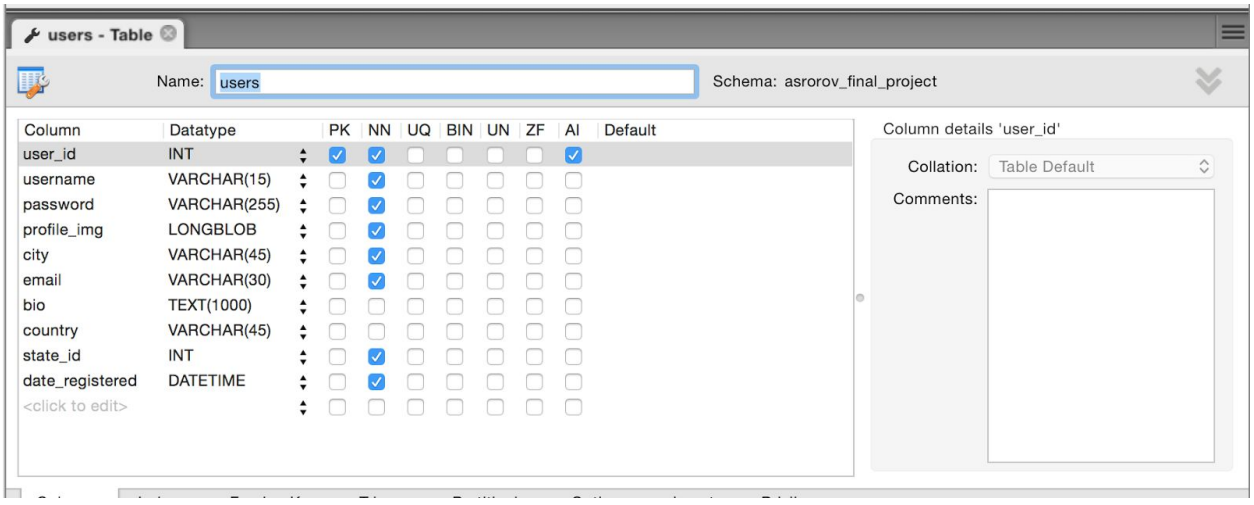





Name: posts
Schema: asrorov_final_project
Column | Datatype | PK | NN | UQ | BIN | UN | ZF | AI | Default
post_id | INT
post_img | LONGBLOB
description | TEXT(1000)
geo_lat | FLOAT
geo_long | FLOAT
category_id | INT
video_link | VARCHAR(45)
Column details 'post_id'
Collation: Table Default
Comments:
Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options | Inserts | Privileges


states - Table
Name: states
Schema: asrorov_final_project
Column | Datatype | PK | NN | UQ | BIN | UN | ZF | AI | Default
state_id | INT
state | VARCHAR(2)
<click to edit>
Column details 'state_id'
Collation: Table Default
Comments:
Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options | Inserts | Privileges


tags - Table
Name: tags
Schema: asrorov_final_project
Column | Datatype | PK | NN | UQ | BIN | UN | ZF | AI | Default
tag_id | INT
tag | VARCHAR(45)
<click to edit>
Column details 'tag_id'
Collation: Table Default
Comments:
Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options | Inserts | Privileges


users - Table
Name: users
Schema: asrorov_final_project
Column | Datatype | PK | NN | UQ | BIN | UN | ZF | AI | Default
user_id | INT
username | VARCHAR(15)
password | VARCHAR(255)
profile_img | LONGBLOB
city | VARCHAR(45)
email | VARCHAR(30)
bio | TEXT(1000)
country | VARCHAR(45)
state_id | INT
date_registered | DATETIME
<click to edit>
Column details 'user_id'
Collation: Table Default
Comments:

Requirements Report

1. Customization:
    a. The level of customization is higher than the assignments:
        i. The user can update, add, delete various types of data that belongs to the user. The user can change his or her content anytime before he/she closes the browser, thanks to sessions.
        ii. The User Interface allows the user to easily come from one page to another without having to refresh the page.
        iii. The content is sorted into different categories and tags. Once the user clicks on specific tag or category, the user will be directed to the set of content that belongs to this particular category/tag.
    b. Used knowledge that was not discussed in this class such as integration of media in php and communication between JavaScript and PHP to display the map provided by Google Maps API.
2. Database
    a. The design and implementation of the database were designed by me.
    b. I populated data with users, posts, and location using the app itself.
    c. As shown above, there are multiple

3. Site Content
    a. Different kinds of graphics, text, and a map is added to the app. For example, there is a tutorial page that explains how to use the app, and what each page contains and what functions in each page.
    b. The website doesn't only contain only database info. It also contains some icons, explanations, images, and videos.

4. Site Design
    a. Each page contains the same kind of template but different structure. Also, the website is responsive with help of the Bootstrap 3 framework.

5. Administrative Functionality and Basic Security
    a. The users are able to edit, delete, and add content in *list_of_items.php* page. A user is able to see the content of other people. He or she can query different types of post by clicking on "Category" or "Mood" link below each post in the *feed.php* page. For example, if you click on "Category: hiking" it will show you all the posts where the category is hiking. Or, if you click on "Mood:funny", it will show you all the posts with the mood funny.

b. All MySQL queries from PHP use OOP MySQLi interface. It can be seen in any page.
c. Not all users are able to perform special operations for the content that do not belong to them. Each user is administrator of his/her content, and is able to delete/edit the content. It was hard to implement it since each user has unique privileges for his/her own content. I wanted to make it seem more like a social network rather than a website with administrator and clients where each user is client and administrator at the same time. As a client, each user can view the users' profiles and content on feeds page, and as administrators they can add, edit, and delete his or her own content.
d. The passwords are hashed before stored in the database for security. Also, if the user tries to go to login page without logging out (enter path in the browser), he or she will be directed back to the feed.php page for security.

6. Extras:
    a. Sessions. Used sessions to display and manipulate content based on the user.
    b. Membership system. Anyone is allowed to upload media such as images and link to the videos.
    c. File upload. The user is able to upload images for both posts and profile.
    d. 3rd party API (Google Maps). With help of Google Maps API, I render the location of specific post indicated by the user after getting the geolocation of the city they provided.
    e. Meaningful confirmation email with the username and password if sign-up or change of password is successful. (might not be extra, but just in case).

Performance Notes

Some pages take some time to load. I guess the problem is with the image processing and the time it takes to load the images from database.