



EAST WEST UNIVERSITY

CSE464: Advance Database Systems

[Fall 2022]

Section: 01

Project SRS



Blogger Spirit

Submitted by:

Student ID	Student Name
2019-1-60-256	Md. Sabik Alam Rahat
2018-3-60-052	Sourav Saha
2018-3-60-108	Md. Abid Hasan Rafe

Project File: https://github.com/sabikrahat/blogger_spirit

Blogger Spirit

Introduction:

Blogger Spirit is a web-based application where a person can easily write his or her blog. For each post 10 points will be reduced. Users can also transfer point from one account to another. User can also see his transferred points and received points. Users can also give feedback and also see other feedbacks

Technologies:

We had use **Python (Django)** to develop this system because python is a very secure language and easy to use. For designing, we had use **HTML, CSS, JavaScript, Bootstrap**. We had use **Xampp Control Panel** as a main server because it is a locally created virtual server in which we can easily create **Apache MySQL** server for our project. Moreover, Xampp is free of cost and easy to use. We had also use **mongo dB** database to store users' feedbacks. Besides, we had use git version control to track our workflow neat and clearly.

Trigger Commands:

```
-- insert into post table
-- 1
DELIMITER $$
CREATE OR REPLACE TRIGGER check_user_points_while_post
BEFORE
    INSERT ON app_posts
FOR EACH ROW
BEGIN
    DECLARE points INT;
    SELECT point INTO points FROM app_users WHERE id = NEW.publisherId;
    IF points < 10 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'You don't have enough point to post';
    END IF;
END $$
DELIMITER ;

-- 2
DELIMITER $$
CREATE OR REPLACE TRIGGER update_user_point_after_post
AFTER
    INSERT ON app_posts
FOR EACH ROW BEGIN
    UPDATE app_users SET point = point - 10 WHERE id = NEW.publisherId;
END $$
DELIMITER ;

-- insert into transfer point table
DELIMITER $$
CREATE OR REPLACE TRIGGER check_user_points_while_transferring
BEFORE
    INSERT ON app_transfer_point
FOR EACH ROW
BEGIN
    DECLARE points INT;
    SELECT point INTO points FROM app_users WHERE email = NEW.senderEmail;
    IF points < NEW.point THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'You don't have enough point to post';
    END IF;
END $$
DELIMITER ;
```

Fig: Trigger Commands

Procedure Commands:

```
-- Procedure

CREATE OR REPLACE PROCEDURE getPosts()
NOT DETERMINISTIC CONTAINS SQL
SQL SECURITY DEFINER
SELECT ap.title, ap.description, ap.img as postImg, ap.created_at as postedAt, au.name, au.email,
au.img as userImg
FROM app_posts ap, app_users au
WHERE ap.publisherId = au.id
ORDER BY ap.id DESC;

CREATE OR REPLACE PROCEDURE getUserPosts(IN userId BIGINT)
NOT DETERMINISTIC CONTAINS SQL
SQL SECURITY DEFINER
SELECT ap.title, ap.description, ap.img as postImg, ap.created_at as postedAt, au.name, au.email,
au.img as userImg
FROM app_posts ap, app_users au
WHERE ap.publisherId = au.id and ap.publisherId = userId
ORDER BY ap.id DESC;

CREATE OR REPLACE PROCEDURE getPostById(IN postId BIGINT)
NOT DETERMINISTIC CONTAINS SQL
SQL SECURITY DEFINER
SELECT ap.title, ap.description, ap.img as postImg, ap.created_at as postedAt, au.name, au.email,
au.img as userImg
FROM app_posts ap, app_users au
WHERE ap.publisherId = au.id and ap.id != postId;
```

Fig: Procedure Commands

Functions Commands:

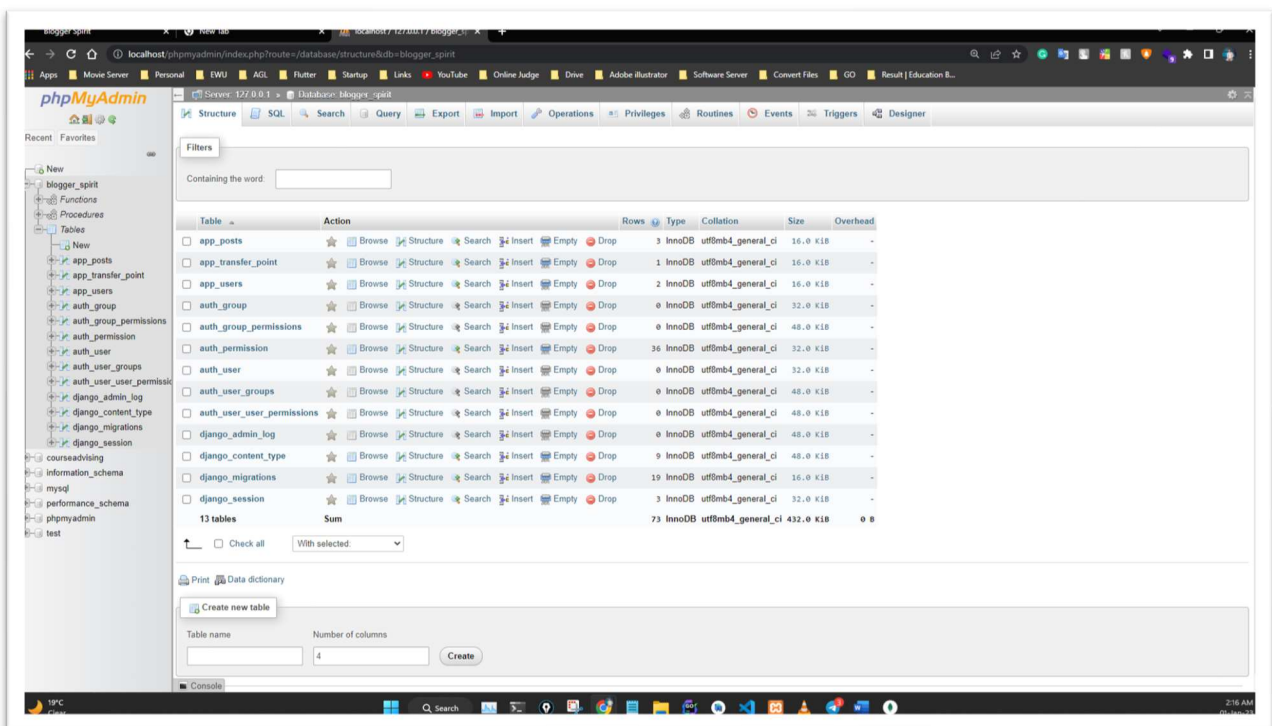
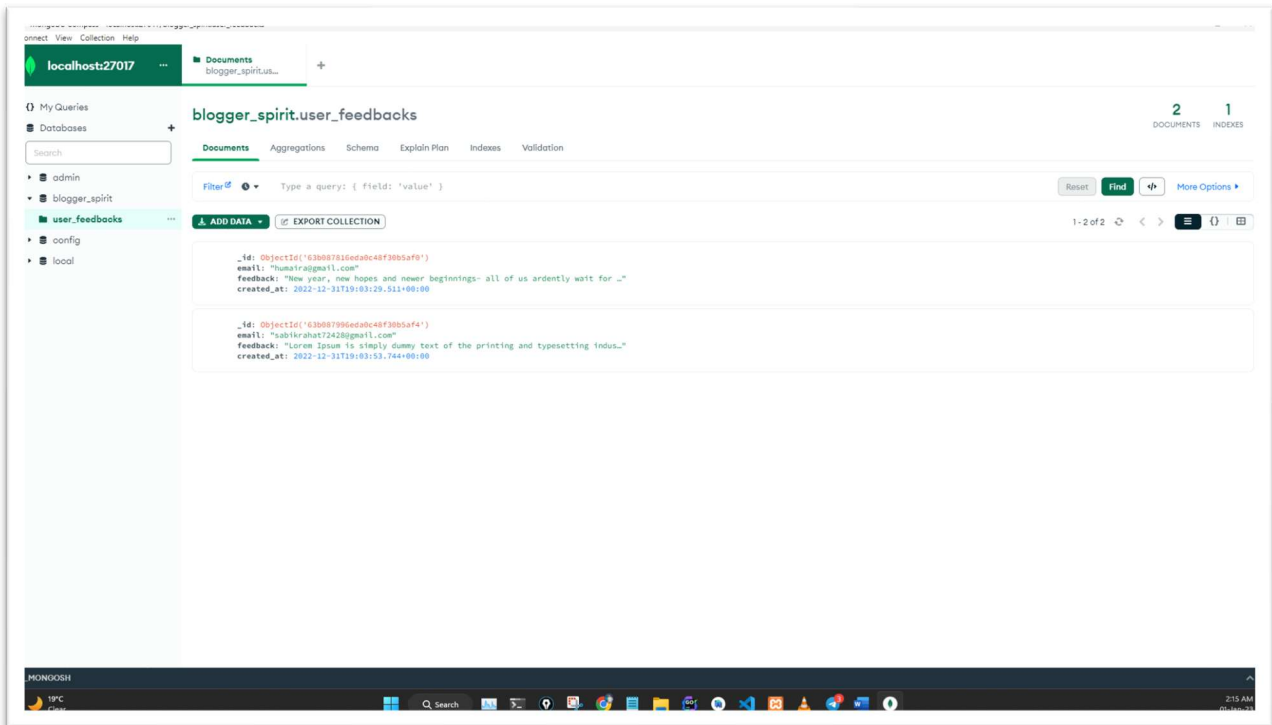
```
-- Function

-- 1 to get total point sendd by user
CREATE OR REPLACE FUNCTION getSendedPoint(userEmail VARCHAR(40))
RETURNS INT NOT DETERMINISTIC CONTAINS SQL
SQL SECURITY DEFINER
RETURN (
    SELECT SUM(point) as total_sended_point
    FROM app_transfer_point
    WHERE senderEmail = userEmail
);

-- 2 to get total point received by user
CREATE OR REPLACE FUNCTION getReceivedPoint(userEmail VARCHAR(40))
RETURNS INT NOT DETERMINISTIC CONTAINS SQL
SQL SECURITY DEFINER
RETURN (
    SELECT SUM(point) as total_received_point
    FROM app_transfer_point
    WHERE receiverEmail = userEmail
);
```

Fig: Functions Commands

Databases:



GUI Screenshots:

