

WAPH-Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Mini Facebook Application

Team members

1. Venaganti Charan Sai , venagaci@mail.uc.edu
2. Sai Abhishek Avusula , avusulsk@mail.uc.edu
3. Ramya Padidala , padidara@mail.uc.edu
4. Varsha Thalishetti , thaliswa@mail.uc.edu

Project Management Information

Source code repository (private access): <https://github.com/waph-team08/waph-teamproject>

Project homepage (public): <https://waph-team08.github.io/>

Revision History

Date	Version	Description
04/16/2024	0.3	users can add post and add comments and delete them

Overview

Sprint 2 revolves around establishing the foundation for user-generated content and interactions. Key tasks include designing and implementing the database structure, enabling logged-in users to create new posts, add comments to any post, and manage their own posts through editing or deletion. However, users will not have the ability to modify posts created by others, ensuring content ownership. This sprint aims to provide a platform for users to contribute content and engage in discussions within the application.

User Interface

Implementation

For our WAPH Team Project, we're adopting an agile development approach, aiming for iterative and collaborative progress. We're utilizing a full-stack development model, employing JavaScript for both frontend and backend development

to ensure consistency and streamline communication between layers. For the database, we're implementing a relational database schema using MySQL or PostgreSQL, facilitating efficient data management and retrieval. Throughout the process, we'll maintain clear documentation and regular communication via platforms like Microsoft Teams to facilitate effective collaboration and progress tracking.

Security analysis

Prepared Statements: Protect against SQL injection attacks by using prepared statements in database queries.

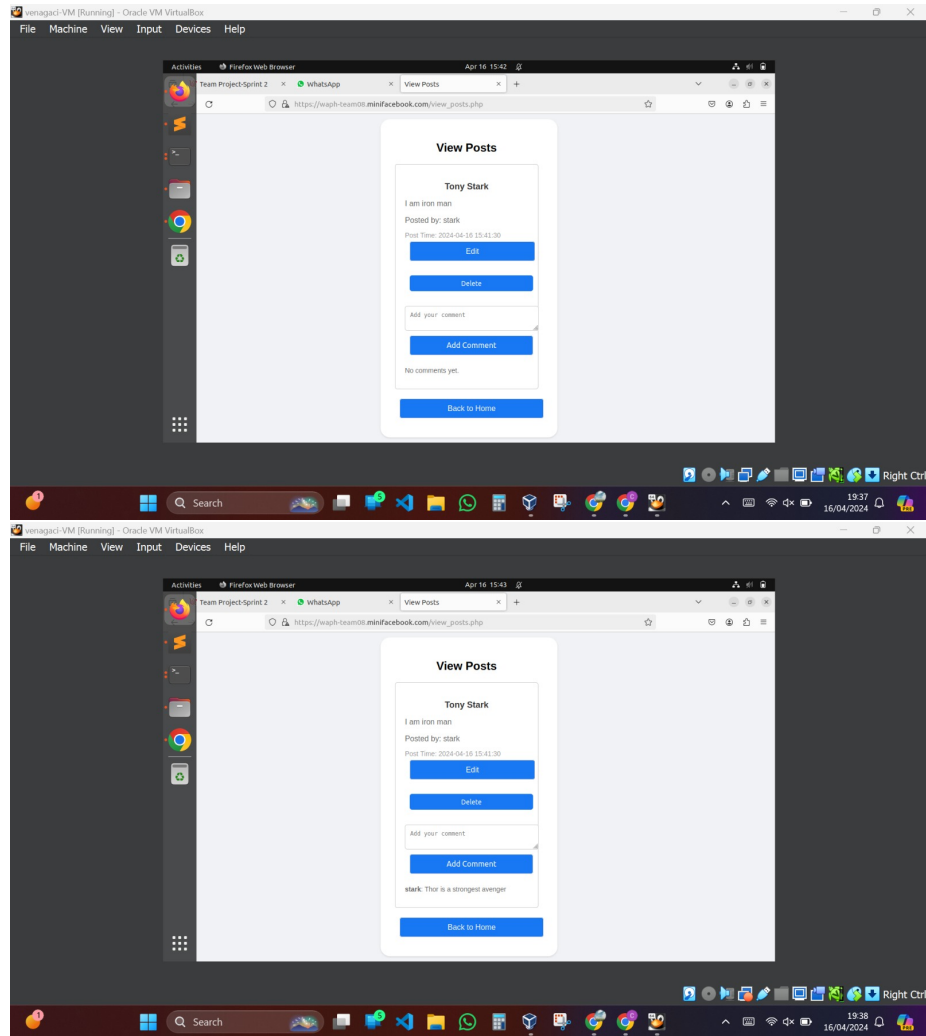
CSRF Token Validation: Prevent CSRF attacks by validating random tokens with each request.

Password Hashing: Securely store passwords with bcrypt hashing to prevent plaintext exposure.

Session Management: Ensure secure session handling with HTTPS, secure cookies, and session regeneration.

Input Validation: Validate and sanitize user input to prevent injection and XSS attacks.

Execution Screenshots



```
venagaci-vm@venagaci ~
mysql> select * from posts;
+-----+-----+-----+-----+-----+
| title | content | posttime | owner | postID |
+-----+-----+-----+-----+-----+
| charan sai headshot | headshot | 2024-04-03 12:42:17 | demo1@gmail.com | 1 |
| headshot | my headshot | 2024-04-03 12:58:01 | demo1@gmail.com | 2 |
| rockstar | John | 2024-04-03 15:45:41 | MALL | 3 |
| king | vkrat kohli | 2024-04-03 15:47:36 | demo1@gmail.com | 4 |
| finisher | dhoni | 2024-04-03 15:56:16 | demo14 | 5 |
| jeddu | ravindra | 2024-04-04 16:24:05 | demo19 | 6 |
| Hello | Hello this is Charam Sai | 2024-04-05 15:16:57 | demo41 | 7 |
| TCS | web application programming and hacking | 2024-04-06 19:03:21 | venagaci | 8 |
| Posti | TCS | 2024-04-06 11:06:47 | sai | 9 |
| wagh-venagaci | Topics in computer systems course is excellent | 2024-04-09 11:20:23 | lenovo | 10 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

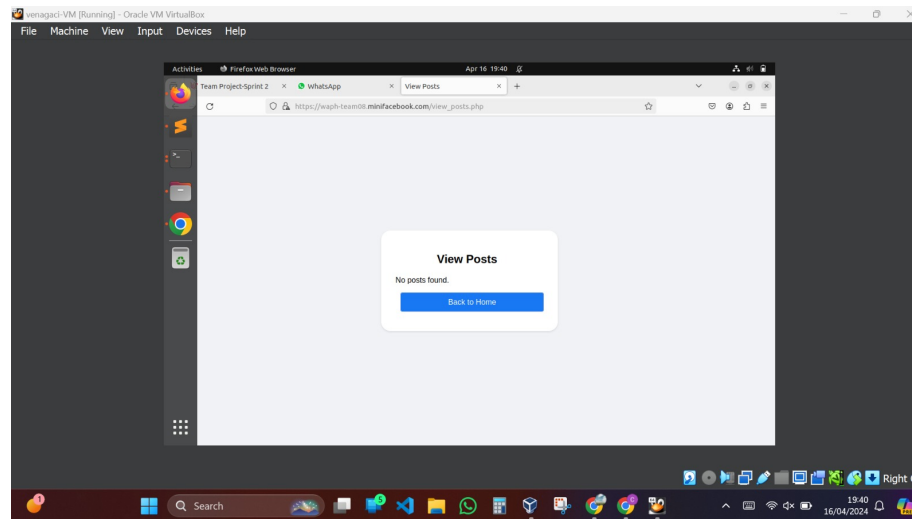
mysql> select * from posts;
+-----+-----+-----+-----+-----+
| title | content | posttime | owner | postID |
+-----+-----+-----+-----+-----+
| charan sai headshot | headshot | 2024-04-03 12:42:17 | demo1@gmail.com | 1 |
| headshot | my headshot | 2024-04-03 12:58:01 | demo1@gmail.com | 2 |
| rockstar | John | 2024-04-03 15:45:41 | MALL | 3 |
| king | vkrat kohli | 2024-04-03 15:47:36 | demo1@gmail.com | 4 |
| finisher | dhoni | 2024-04-03 15:56:16 | demo14 | 5 |
| jeddu | ravindra | 2024-04-04 16:24:05 | demo19 | 6 |
| Hello | Hello this is Charam Sai | 2024-04-05 15:16:57 | demo41 | 7 |
| TCS | web application programming and hacking | 2024-04-06 19:03:21 | venagaci | 8 |
| Posti | TCS | 2024-04-06 11:06:47 | sai | 9 |
| wagh-venagaci | Topics in computer systems course is excellent | 2024-04-09 11:20:23 | lenovo | 10 |
| God of thunder | worthythor | 2024-04-10 20:07:52 | chris | 11 |
| Tony Stark | I am Iron man | 2024-04-16 15:41:30 | stark | 12 |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql>
```

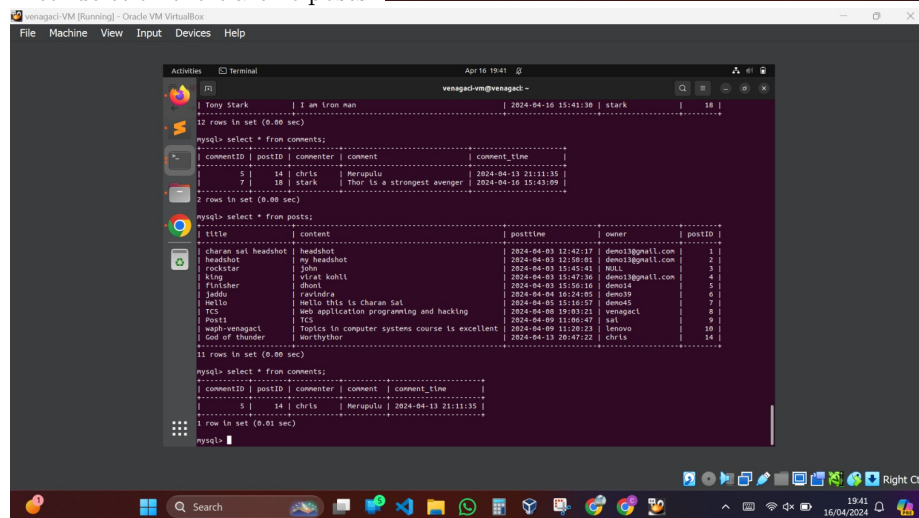
```
venagaci-vm@venagaci ~
mysql> select * from posts;
+-----+-----+-----+-----+-----+
| title | content | posttime | owner | postID |
+-----+-----+-----+-----+-----+
| charan sai headshot | headshot | 2024-04-03 12:42:17 | demo1@gmail.com | 1 |
| headshot | my headshot | 2024-04-03 12:58:01 | demo1@gmail.com | 2 |
| rockstar | John | 2024-04-03 15:45:41 | MALL | 3 |
| king | vkrat kohli | 2024-04-03 15:47:36 | demo1@gmail.com | 4 |
| finisher | dhoni | 2024-04-03 15:56:16 | demo14 | 5 |
| jeddu | ravindra | 2024-04-04 16:24:05 | demo19 | 6 |
| Hello | Hello this is Charam Sai | 2024-04-05 15:16:57 | demo41 | 7 |
| TCS | web application programming and hacking | 2024-04-06 19:03:21 | venagaci | 8 |
| Posti | TCS | 2024-04-06 11:06:47 | sai | 9 |
| wagh-venagaci | Topics in computer systems course is excellent | 2024-04-09 11:20:23 | lenovo | 10 |
| God of thunder | worthythor | 2024-04-10 20:07:52 | chris | 11 |
| Tony Stark | I am Iron man | 2024-04-16 15:41:30 | stark | 12 |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)

mysql> select * from comments;
+-----+-----+-----+-----+-----+
| commentID | postID | commenter | comment | comment_time |
+-----+-----+-----+-----+-----+
| 5 | 7 | chris | Merupulu | 2024-04-12 21:11:35 |
| 7 | 10 | stark | Thor is a strongest avenger | 2024-04-16 15:43:09 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```



After deletion there are no posts.



Scrum process

Sprint 0

Duration: 19/03/2024-22/03/2024

Overall team Completed Tasks:

1. Create an organization on GitHub with name waph-team08

Added the other members and phung-waph as organization members 2. Member contributions to the team organization:

- Created a private repository for the project, namely waph-teamproject

- Created a public repository to host the team project website, namely, waph-team08.github.io
 - Checkedout the waph-teamproject repository and added README.md template
 - Checkedout the waph-team08.github.io repository and added index.html template
 - Checkedout the waph-teamproject and waph-team08.github.io repositories, added/revised the code, README.md, and index.html files accordingly; committed/pushed the changes
3. Created team SSL key/certificate and setup HTTPS and team local domain name.
 4. Design and setup the team database
 5. Copy the code skeleton from Lab 3 to the team repository waph-teamproject, revise the code based on Lab 4 requirements test and committed the code together with README.md and index.html for Sprint 0 submission.

Sprint 1

Duration: 03/25/2024-03/31/2024

Completed Tasks:

1. User Registration and login
2. change password
3. edit profile
4. view posts

Contributions:

1. Member 1, 4 commits, 1 hours, contributed in xxx
2. Member 2, 6 commits, 1 hours, contributed in xxx
3. Member 3, 4 commits, 1 hours, contributed in xxx
4. Member 4, 3 commits, 1 hours, contributed in xxx

Sprint 2

Duration: 03/31/2024-04/16/2024

Completed Tasks:

1. Logged-in users can add a new post, and add a comment on any post.
2. Logged-in users can edit (update, delete) their own posts.
3. A user cannot edit posts of other users change password.

Contributions:

1. Member 1, 8 commits, 3 hours, contributed in view__posts.php, database-data.sql
2. Member 2, 7 commits, 1 hours, contributed in delete_post.php
3. Member 3, 6 commits, 1 hours, contributed in edit_post.php
4. Member 4, 5 commits, 1 hours, contributed in add_comment.php

Appendix

database-data.sql:

```
DROP TABLE IF EXISTS users;
CREATE TABLE users(
    username VARCHAR(50) PRIMARY KEY,
    password VARCHAR(100) NOT NULL,
    additional_email VARCHAR(100),
    phone VARCHAR(20)
    name VARCHAR(20)
);

INSERT INTO users(username, password, additional_email, phone) VALUES ('waph_team08', md5('P
INSERT INTO users(username, password, additional_email, phone) VALUES ('test', md5('test'),

DROP TABLE IF EXISTS posts;
CREATE TABLE posts(
    postID VARCHAR(100) INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    content VARCHAR(100),
    posttime VARCHAR(100),
    owner VARCHAR(100),
    FOREIGN KEY (owner) REFERENCES users (username) ON DELETE CASCADE
);

CREATE TABLE comments (
    commentID INT AUTO_INCREMENT PRIMARY KEY,
    postID INT,
    commenter VARCHAR(50),
    comment TEXT,
    comment_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (postID) REFERENCES posts(postID),
    FOREIGN KEY (commenter) REFERENCES users(username)
);
```