Social Media Platform Programming

Overview

In this assignment, you will be developing a basic text-based social media platform using Python. The platform will allow users to create posts, view a timeline, like posts, and comment on posts. You'll write several functions to accomplish these tasks.

Function 1: `initializePlatform()` - 2 pts

Write a function called `initializePlatform()` that doesn't take any arguments and returns an empty list representing the initial state of the platform.

Function 2: `createPost(platform, content)` - 4 pts

Write a function called `createPost()` that takes the current platform and the content of the post as arguments. The function should create a new post with the content, initialize the likes to 0, and initialize an empty list for comments. The new post should be added to the platform. The function should return the updated platform.

Function 3: `viewTimeline(platform)` - 4 pts

Write a function called `viewTimeline()` that takes the current platform as an argument and prints each post in the platform. Each post should display its content, the number of likes, and the number of comments. Use appropriate formatting to separate posts.

Function 4: `likePost(platform, postIndex)` - 3 pts

Write a function called `likePost()` that takes the current platform and the index of the post to be liked as arguments. The function should increment the number of likes for the specified post by 1. If the post index is invalid, the function should print an error message. The function should return the updated platform.

Function 5: `commentOnPost(platform, postIndex, comment)` - 3 pts

Write a function called `commentOnPost()` that takes the current platform, the index of the post to be commented on, and the comment as arguments. The function should add the comment to the specified post's list of comments. If the post index is invalid, the function should print an error message. The function should return the updated platform.

Function 6: `startPlatform()` - 9 pts

Write a function called `startPlatform()` that runs the main loop for the social media platform.

Steps:

a. **Setup**: Print a welcome message and initialize the platform. b. **Main Loop**: Create a while-loop that presents a menu with options to create a post, view the timeline, like a post, comment on a post, or exit the platform. The loop should continue until the user chooses to exit. c. **Create Post**: Prompt the user to enter the content of the post and call `createPost()` to add the post to the platform. d. **View Timeline**: Call `viewTimeline()` to display all posts on the platform. e. **Like Post**: Prompt the user to enter the index of the post to like and call `likePost()` to increment the likes for the specified post. f. **Comment on Post**: Prompt the user to enter the index of the post to comment on and the comment itself, then call `commentOnPost()` to add the comment to the specified post. g. **Exit**: If the user chooses to exit, print a goodbye message and end the loop.

Example Interaction

1. **Start**: The user launches the platform and sees a welcome message.
2. **Create Post**: The user selects the option to create a new post, enters the content, and the post is added to the platform.
3. **View Timeline**: The user selects the option to view the timeline and sees all the posts with their likes and comments.
4. **Like Post**: The user selects the option to like a post, enters the post index, and the likes for that post are incremented.
5. **Comment on Post**: The user selects the option to comment on a post, enters the post index and the comment, and the comment is added to the post.
6. **Exit**: The user selects the option to exit, sees a goodbye message, and the platform closes.

Notes

- Ensure the platform correctly handles invalid inputs and provides clear feedback to the user.
- The timeline should be updated to reflect the current state of posts, likes, and comments after each action.
- The main loop should continue running until the user chooses to exit the platform.