

Task-5

Blog Platform:

Build a simple blog platform where users can create, read, update, and delete posts, with comments and user profiles.

CODE:

post.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Blog Platform</title>
  <link rel="stylesheet" href="blogg.css">
</head>
<body>
  <header>
    <h1>Simple Blog Platform</h1>
    <nav>
      <ul>
        <li><a href="#register">Register</a></li>
        <li><a href="#login">Login</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="contact">Contact</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <!-- display posts -->
    <section id="posts">
      <h2>Posts</h2>
      <ul id="post-list"></ul>
```

```

</section>

<!-- create post form -->
<section id="create-post">
  <h2>Create Post</h2>
  <form id="create-post-form">
    <input type="text" id="post-title" placeholder="Title">
    <textarea id="post-content"
placeholder="Content"></textarea>
    <button type="submit">Create Post</button>
  </form>
</section>

<!-- post details -->
<section id="post-details"></section>

<!-- comments -->
<section id="comments"></section>

<!-- user profile -->
<section id="user-profile"></section>
</main>

<!-- scripts -->
<script src="script.js"></script>
</body>
</html>

```

Blogg.css:

```

body {
  font-family: Calibri, sans-serif;
}

header {

```

```
    background-color: #333;
    color: #fff;
    padding: 1em;
    text-align: center;
}
```

```
nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
}
```

```
nav li {
    display: inline-block;
    margin-right: 20px;
}
```

```
nav a {
    color: #fff;
    text-decoration: none;
}
```

```
#posts {
    max-width: 800px;
    margin: 60px auto;
}
```

```
#post-list {
    list-style: none;
    margin: 0;
    padding: 0;
}
```

```
#post-list li {
```

```
    border-bottom: 1px solid #ddd;
}

#post-list li:last-child {
    border-bottom: none;
}

#post-list li:hover {
    background-color: #f5f5f5;
}

#create-post {
    max-width: 800px;
    margin: 40px auto;
}

#create-post-form {
    width: 100%;
}

#create-post-form input[type="text"], #create-post-form
textarea {
    width: 100%;
}

#create-post-form button[type="submit"] {
    background-color: #333;
    color: #fff;
}

#create-post-form button[type="submit"]:hover {
    background-color: #444;
}
```

blog.js

```
// global variables
```

```
const posts = [];
const users = [];

// register user
document.getElementById('register').addEventListener('click', ()
=> {
    const username = prompt('Enter your username:');
    const password = prompt('Enter your password:');
    users.push({ username, password });
});

// login user
document.getElementById('login').addEventListener('click', () =>
{
    const username = prompt('Enter your username:');
    const password = prompt('Enter your password:');

    for (const user of users) {
        if (user.username === username && user.password ===
password) {
            alert(`Welcome, ${username}!`);
            return;
        }
    }
});

// create post
document.getElementById('create-post-form').addEventListener('su
bmit', (e) => {
    e.preventDefault();

    const title =
document.getElementById('post-title').value.trim();
```

```
const content =
document.getElementById('post-content').value.trim();

if (title && content) {
  const newPost = { title, content };
  posts.push(newPost);

  // update post list
  const postListHTML = '';
  for (const post of posts) {
    postListHTML += `<li>${post.title}</li>`;
  }
  document.getElementById('post-list').innerHTML =
postListHTML;

  // reset form
  document.getElementById('create-post-form').reset();

  alert(`Post created successfully!`);

  // show post details
  document.getElementById('post-details').innerHTML = `
    <h2>${newPost.title}</h2>
    <p>${newPost.content}</p>`;

  // show comments
  document.getElementById('comments').innerHTML = '';

  // show user profile
  document.getElementById('user-profile').innerHTML = `
    <h2>${username}</h2>`;

  return false;
});
```

```

// update post
document.getElementById('post-list').addEventListener('click',
(e) => {
  if (e.target.tagName === 'LI') {
    const postId = e.target.textContent.trim();
    for (const post of posts) {
      if (post.title === postId) {
        // update post content
        const newContent = prompt(`Enter new content for
"${postId}":`);
        if (newContent) {
          post.content = newContent;
          // update post list
          const postListHTML = '';
          for (const post of posts) {
            postListHTML += `<li>${post.title}</li>`;
          }
          document.getElementById('post-list').innerHTML =
postListHTML;

          alert(`Post updated successfully!`);
        }
      }
    }
  }
});

```

```

// delete post
document.getElementById('post-list').addEventListener('click',
(e) => {
  if (e.target.tagName === 'LI') {
    const postId = e.target.textContent.trim();
    for (const i in posts) {
      if (posts[i].title === postId) {

```

```
posts.splice(i, 1);  
// update post list  
const postListHTML = '';  
for (const post of posts) {  
    postListHTML += `<li>${post.title}</li>`;  
}  
document.getElementById('post-list').innerHTML =  
postListHTML;  
  
    alert(`Post deleted successfully!`);  
}  
}  
));
```

OUTPUT:

