**Practice Guide for Exam Questions**

**HTML Tasks**

**Task 1a: Create an HTML Form for Tweets**

<form action="/submitTweet" method="post">

<label for="tweetContent">Tweet:</label>

<textarea id="tweetContent" name="content" rows="4" cols="50" placeholder="Write your tweet here..."></textarea>

<button type="submit">Submit</button>

</form>

**Task 1b: Display a Tweet**

<div class="tweet">

<img src="profile.jpg" alt="Profile Picture" class="profile-pic">

<div class="content">

<p>Tweet content goes here...</p>

<span class="timestamp">Posted on: Jan 12, 2025</span>

</div>

</div>

**Task 1c: Save Button with FontAwesome Icon**

**Task 1d: Prevent Cross-Site Scripting (XSS)**

To avoid XSS, sanitize all user input and output. For example:

* Use server-side libraries to escape special characters (e.g., &, <, >).
* In JavaScript, use textContent instead of innerHTML to insert user-generated content.
* Use Content Security Policy (CSP) headers to limit where scripts can execute.

**CSS Tasks**

**Task 2a: Black Background and White Text**

body {

background-color: black;

color: white;

}

**Task 2b: Inline CSS for Grayscale Background and Red Text**

<p style="background-color: gray; color: red;">Style me!</p>

**Task 2c: CSS to Override Font Size**

#user.avatar {

font-size: 32px !important;

}

**Task 2d: Move Element to Top of Page**

.element-to-move {

position: absolute;

top: 0;

left: 0;

}

**Task 2e: Equal Box Sizes**

.box {

padding: 8px;

border: 1px solid black;

box-sizing: border-box;

width: 200px;

height: 100px;

}

**JavaScript Tasks**

**Task 3a: Favorite Button with Toggle Indicator**

<button id="favoriteBtn" onclick="toggleFavorite(this)">Favorite</button>

<script>

function toggleFavorite(button) {

button.style.backgroundColor = button.style.backgroundColor === 'yellow' ? '' : 'yellow';

}

</script>

**Task 3b: Report Button with Counter**

<div class="tweet" style="padding: 10px;">

<button class="report-btn">Report</button>

</div>

<script>

document.querySelector('.report-btn').addEventListener('click', function () {

this.closest('.tweet').style.backgroundColor = 'red';

});

</script>

**SQL Tasks**

**Task 4a: Insert a New User**

INSERT INTO User (firstname, lastname, email)

VALUES ('Anders', 'And', 'anders.and@example.com');

**Task 4b: Update User Name**

UPDATE User

SET firstname = 'Mickey', lastname = 'Mouse'

WHERE firstname = 'Anders' AND lastname = 'And';

**Task 4c: Rename Table**

RENAME TABLE User TO Users;

**C#/.NET MVC Tasks**

**Task 5a: JavaScript AJAX Function for Tweets**

function createTweet(content) {

fetch('/TweetsController/Index?content=' + encodeURIComponent(content))

.then(response => response.json())

.then(data => console.log(data));

}

**Task 5b: JSON Object for Tweet**

{

"id": 1,

"content": "This is a sample tweet",

"author": "John Doe",

"timestamp": "2025-01-12T14:00:00Z"

}

**Task 5c: Favorite Model**

public class Favorite {

public int Id { get; set; }

public int UserId { get; set; }

public int TweetId { get; set; }

public virtual User User { get; set; }

public virtual Tweet Tweet { get; set; }

}

**Task 5d: LINQ Query for Favorites**

public IActionResult Index() {

var userId = GetCurrentUserId();

var favoriteTweets = \_context.Favorites

.Where(f => f.UserId == userId)

.Select(f => f.Tweet)

.ToList();

return View(favoriteTweets);

}

**Task 5e: Display Tweets in View**

@foreach (var tweet in ViewBag.Tweets) {

<div class="tweet">

<p>@tweet.Content</p>

</div>

}