Final Course Website Documentation

05/14/2018

9352A 4:00 – 5:00 TF

Members:

Domaoa, Jeane Cris

Ramos, Tatum Eiffel Dodge

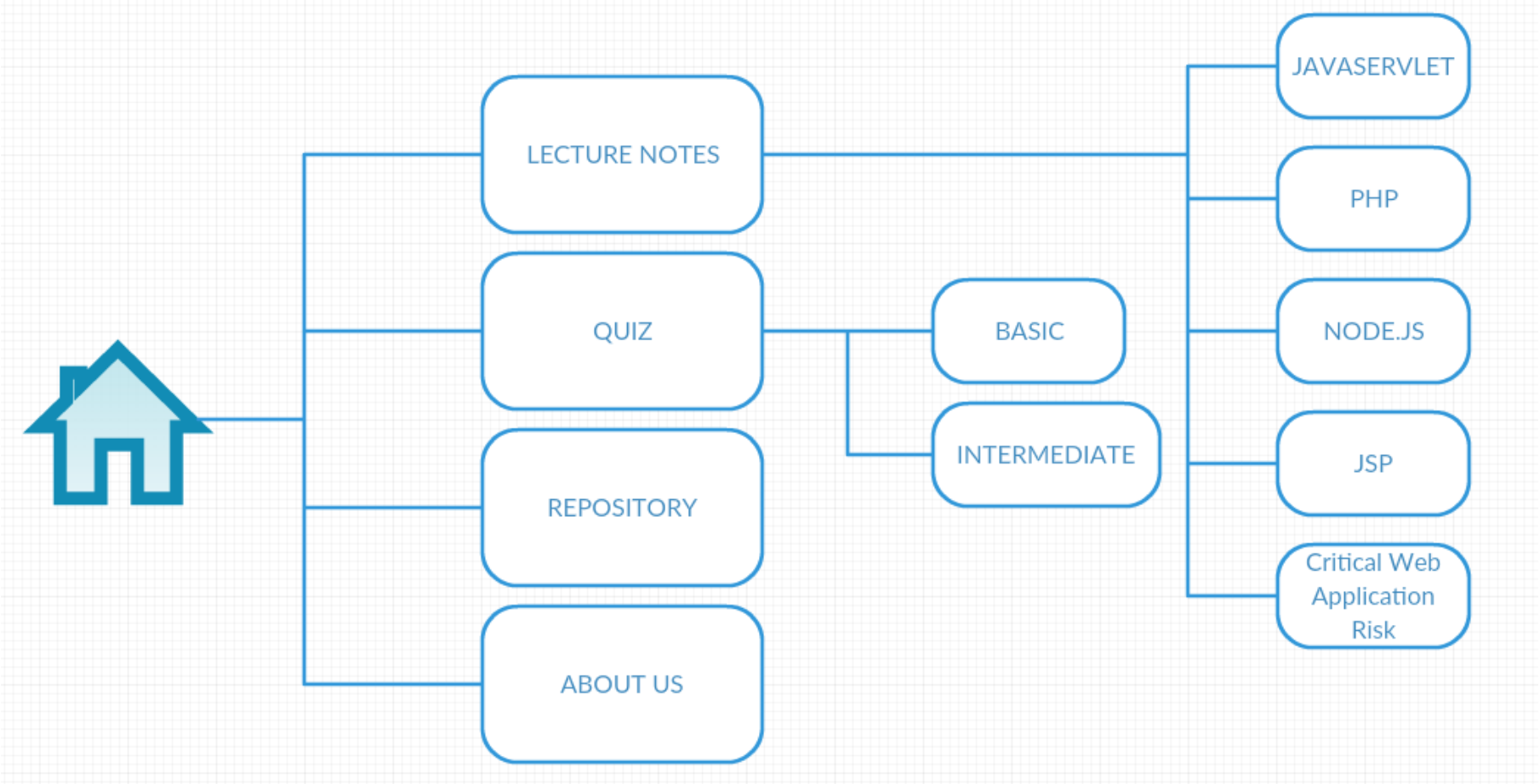
Calines, Carla

Catayao, Roxanne

Dayrit, Aika Vien

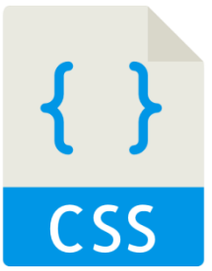
Table of Contents

WEBSITE ARCHITECTURAL DIAGRAM

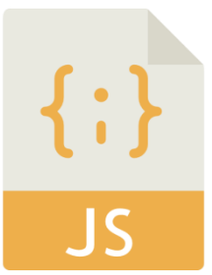


OVERVIEW OF THE TECHNOLOGIES USED IN THE PROJECT

We used HTML to structure the elements of the website, the lecture notes, the quizzes, and the other content. We also used this technology to embed the images used in the website such as the code snippet, developers' profile, and logos.



We used CSS to basically not just to beautify our website but also to give layout to it. We used Cascading Style Sheets to every website we created. We also used frameworks such as bootstrap to easily add styles in our HTML elements just by using class.



We used JavaScript to add additional functionalities to our navigation bar such as adding a style to the navigation bar when the user scrolls into webpages. We also used frameworks like the Bootstrap JS, Popper, and JQuery which are dependencies for the bootstrap CSS to add additional functions to designs.

JSON text files were created to store the different information for the quizzes in the website, the questions and their corresponding choices and correct answer.

We used the PHP technology to administer the quizzes that are implemented on the website. It was used display the quiz items form the JSON text files to the website and it was also used to evaluate the answers of the users against the correct answer to the quiz items.

WEBSITE INTERFACE

Home Page

- Landing Screen
    - The landing screen shows the user the initially transparent navigation bar and a heading as evidenced in the figure below. It also has a button that when clicked, it scrolls the page to the Lecture Notes Section.

Figure 1.0 -Landing Page

- Lecture Notes Section

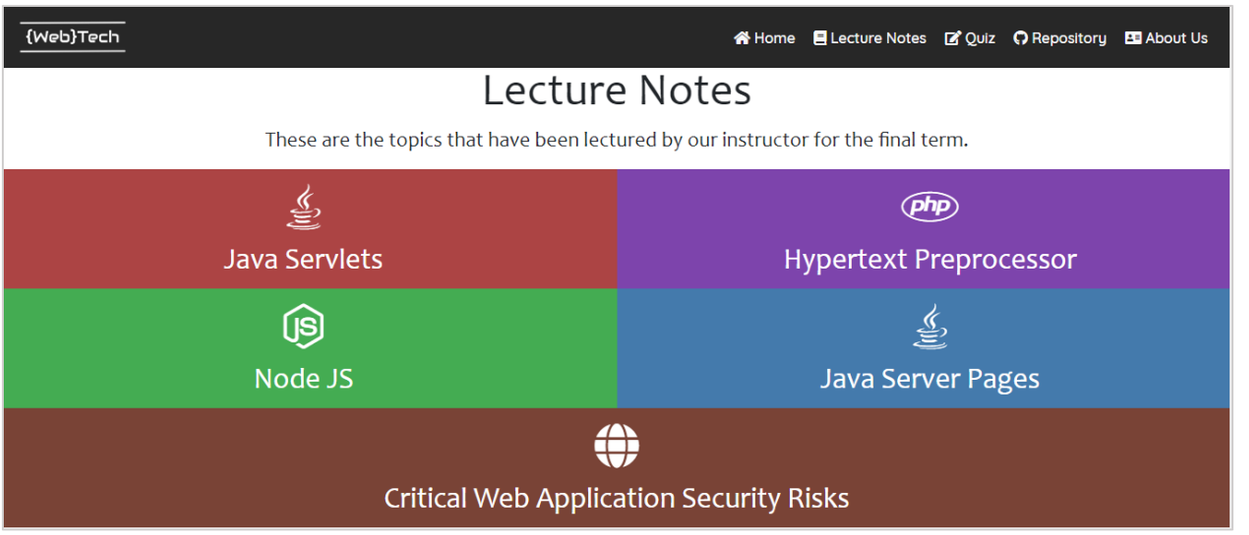


Figure 1.1 -Lecture Notes Section

The section contains the links to the pages containing the lecture topics covered in the Finals period of the course. Clicking on the different topics will redirect you to their corresponding pages:

- Java Servlets

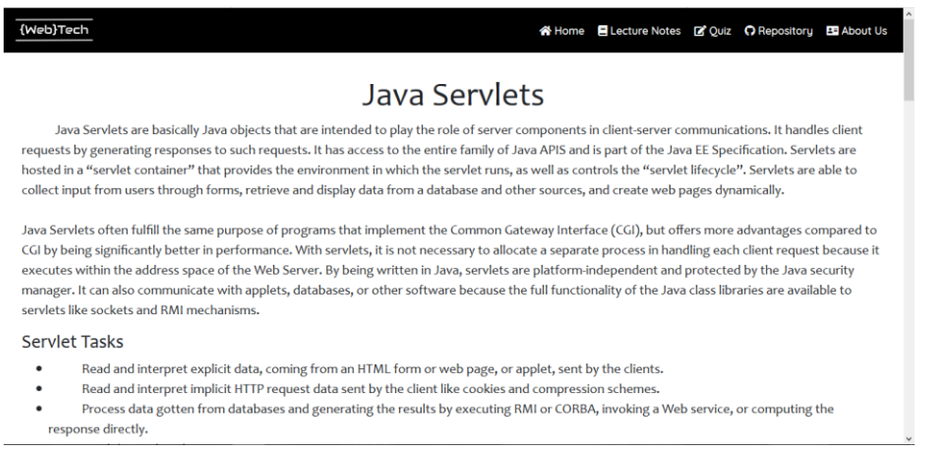


Figure 1.1a -Java Servlets page

- PHP

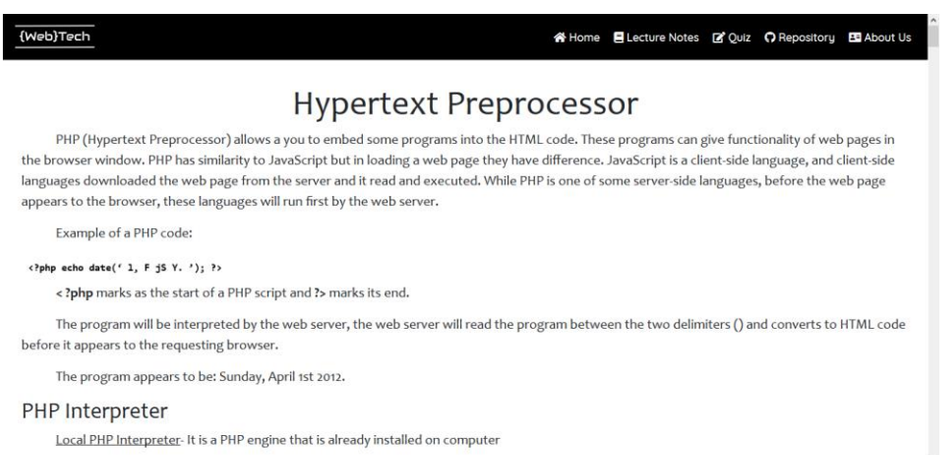


Figure 1.1b – PHP page

- Node JS

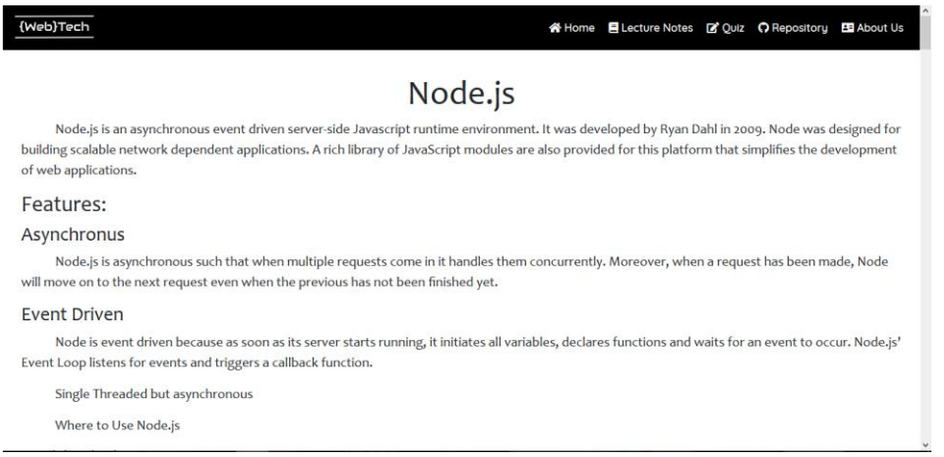


Figure 1.1c -Node.js page

- Critical Web Security Risks

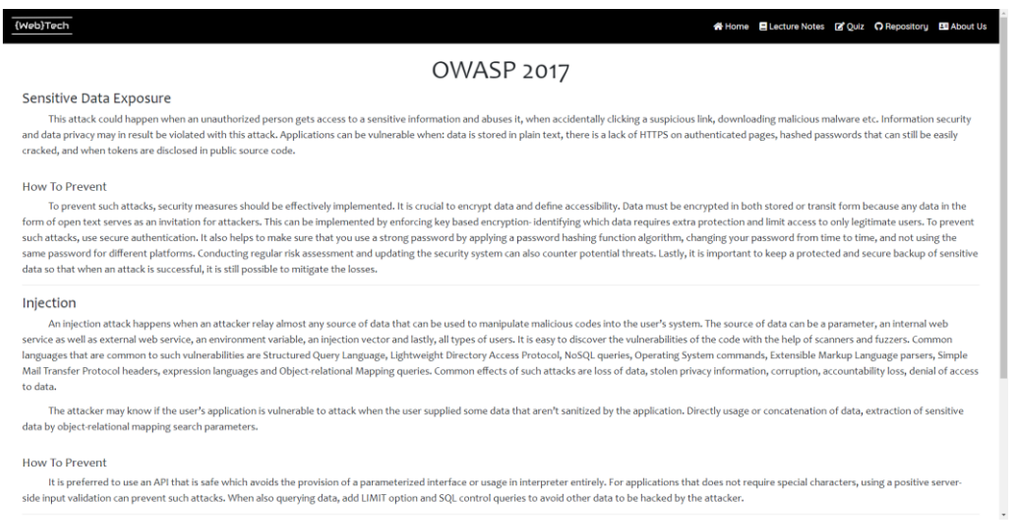


Figure 1.1d –Critical Web Application Risk 2017 page

- Quizzes Section

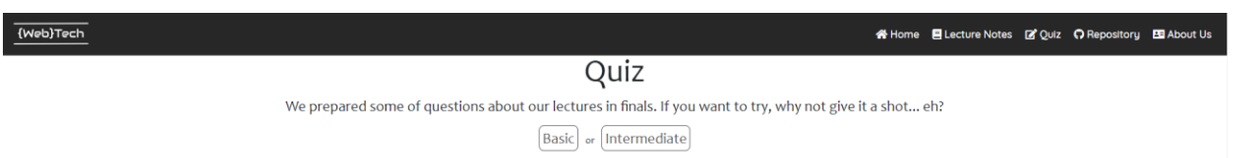


Figure 1.2 – Quiz Section

Clicking on the buttons in the Quizzes section will redirect you to quiz administration page. The user will answer all of the questions and click on the submit button to see his/her score.

- Basic Level Quiz

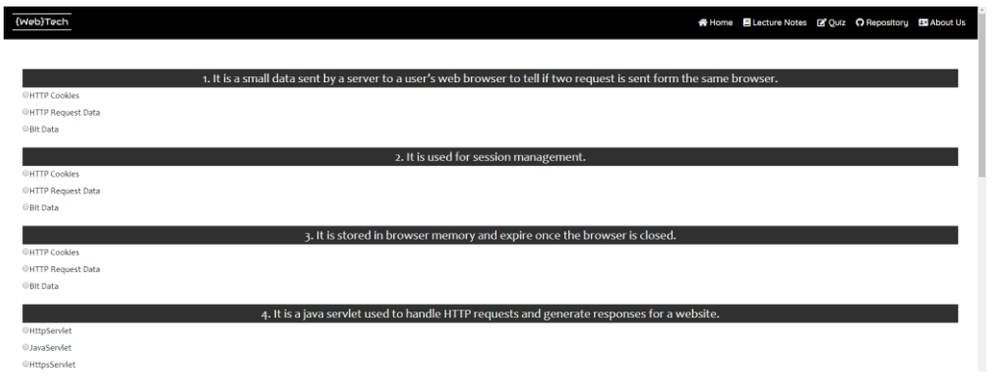


Figure 1.2a -Basic Level Quiz page

- Intermediate Level Quiz

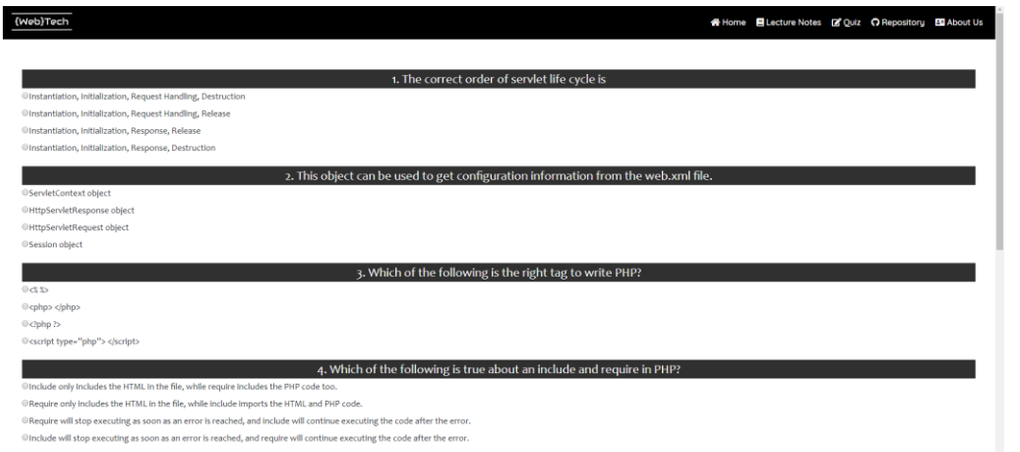


Figure 1.2b -Intermediate Level Quiz page

- Repository Information Section

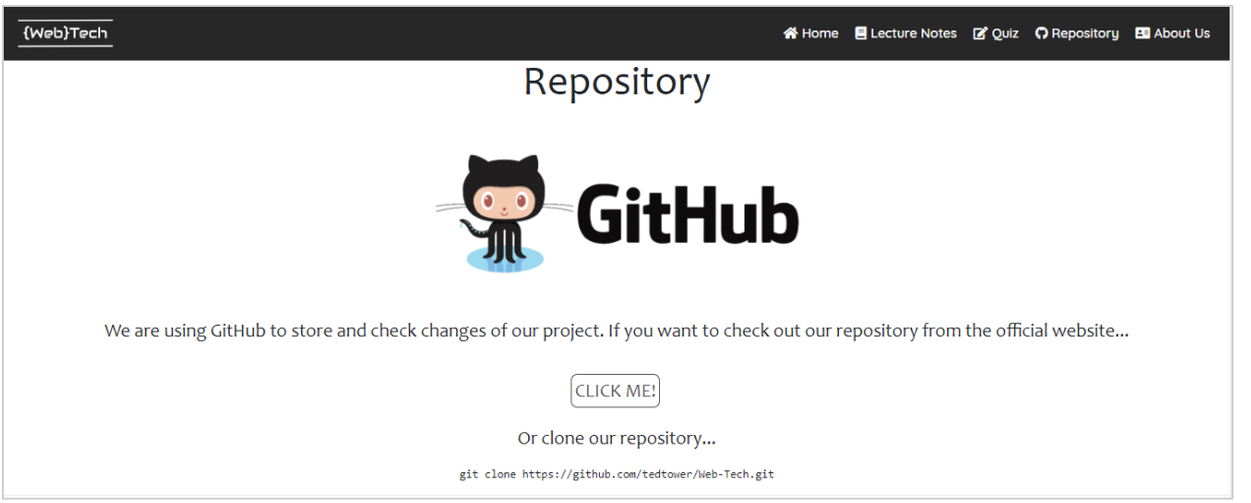


Figure 1.3 -Repository Information section

Clicking on the "CLICK ME!" button will redirect the user to our GitHub repository to the link provided below the button.

- About Us Section

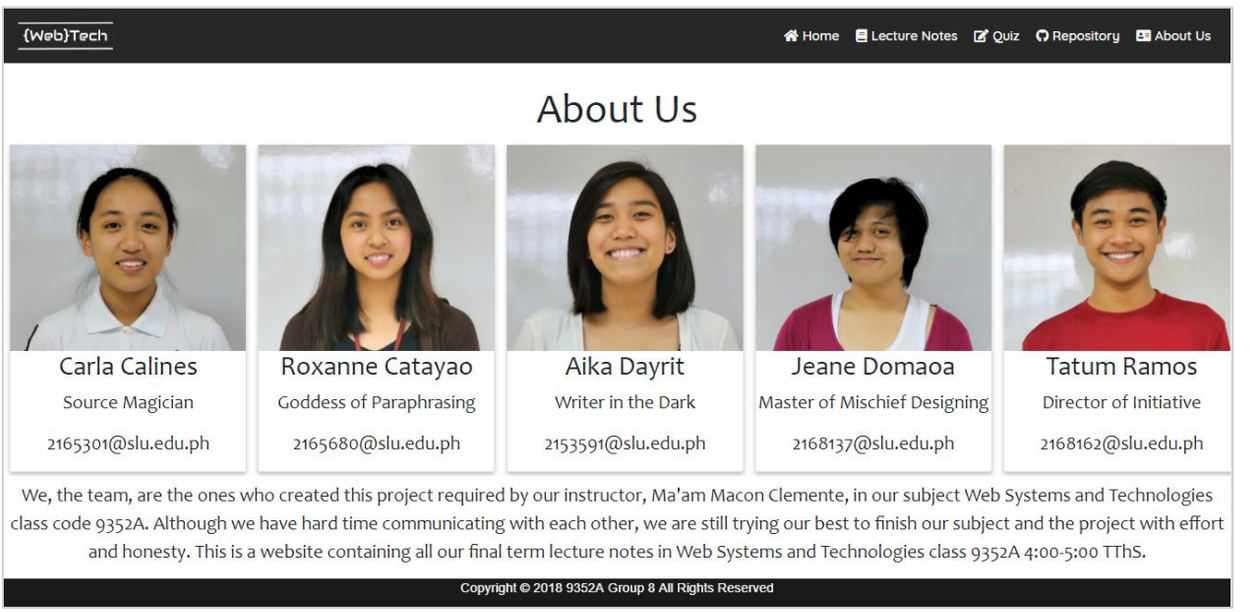


Figure 1.4 About Us section

The About Us section shows the group members of the project. The group member profiles contain the names of the members, a nickname and their contact information. Also included in this section are the description of the website and its purpose.

WEBSITE FEATURES

Responsive

On the screen of a laptop computer the website will look like this

Figure 2.0 -Laptop Computer Screen View

But when the screen size decreases the style of the website changes

Tablet

Figure 2.1 -Tablet screen view

Mobile Phone

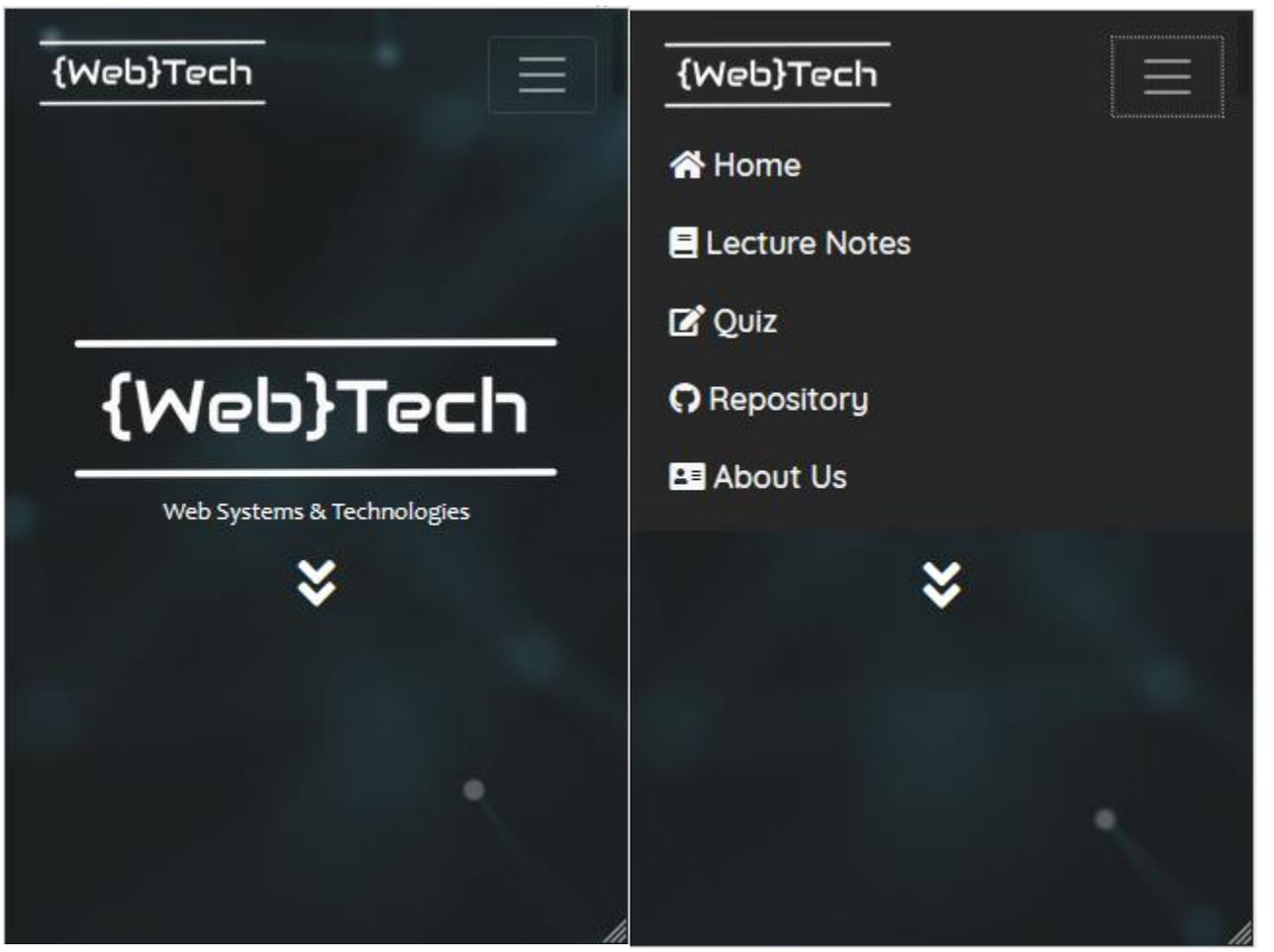


Figure 2.2 -Mobile Phone screen view

@media Rule

With the use of the @media rule we are able to change how different elements on the website will look like depending on the screen width. We optimized the website such that it can be viewable in a more aesthetic way in different devices. As evidenced by the screenshots provided above as the screen width narrows, the style of the navigation bar changes. The navigation links will only appear once you click the “Navigation Collapsible Icon” as shown above.

Server-side Quiz Administration

Instead of doing the quiz administration processes on the client-side, we used the PHP technology. It will first read the contents of the JSON text files that contains the quiz items for each quiz, the Basic and Intermediate quizzes, and it will parse them as associative arrays.

Like so:

```php
<?php
    $json = file_get_contents("../json/questions.json");
    $array = json_decode($json, true);
?>
```

Figure 3.0 – readJson.php

```json
{
    "questions": [
        {
            "question" : "It is a small data sent by a server to a user's web browser to tell if two request is sent form the same browser.",
            "choices" : [
                "HTTP Cookies",
                "HTTP Request Data",
                "Bit Data"
            ],
            "answer" : "a1"
        },
        {
            "question" : "It is used for session management.",
            "choices" : [
                "HTTP Cookies",
                "HTTP Request Data",
                "Bit Data"
            ],
            "answer" : "b1"
        },
        {
            "question" : "It is stored in browser memory and expire once the browser is closed.",
            "choices" : [
                "HTTP Cookies",
                "HTTP Request Data",
                "Bit Data"
            ],
            "answer" : "c1"
        },
        {
            "question" : "It is a java servlet used to handle HTTP requests and generate responses for a website.",
            "choices" : [
                "HttpServlet",
```

Figure 3.1 – questions.php

It will then use the values to be printed on each quiz page. The values inside the JSON file is an object called "questions", as shown above, that contains an array of objects that contains the following values, the question, choices, and the correct answer to the question.

```php
}else {
    if ($array == null) {
        echo "Your array is empty!!!";
    } else {
        $cChar = 0141;

        for ($x = 0; $x < sizeof($array['questions']); $x++) {
            $qNo = $x + 1;
            $choiceChar = chr($cChar);
            echo "\n<p>" . $qNo . ". " . $array['questions'][$x]['question'] . "</p>";

            for ($y = 0; $y < sizeof($array['questions'][$x]['choices']); $y++) {
                $cNo = $y + 1;
                $id = $choiceChar . ($y + 1);
                echo "<input type='radio' name='q$qNo' id='$id' value='$id'><label
                for='$id' style='font-size:17px;'>" . $array['questions'][$x]['choices']
                [$y] . "</label><br>";
            }
            $cChar++;
        }
    }
}
?>
<br>
<div class="butt">
    <input type="submit" value="Submit" name="submit">
</div>
```

Figure 3.2 – quiz.php (Display Quiz)

After the questions have been accomplished it will check the answer of the user. To avoid warnings when submitting the quiz, the user must answer all items. Once the user has clicked the submit button at the bottom of the quiz page, then his/her answers will be evaluated against the correct answers as embedded in the JSON files.

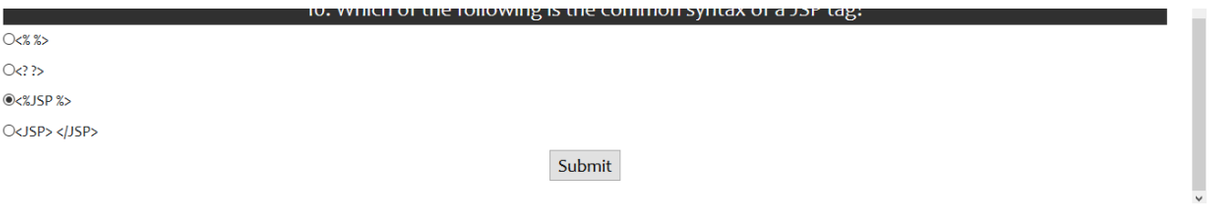


Figure 3.3 – Submit after answering the quiz.

```php
<?php
require "../includes/readJson.php";
if(isset($_POST['submit'])) {
    $array_size = sizeof($array['questions']);
    $correct_answers = [];
    for ($index = 0; $index < $array_size; $index++) {
        array_push($correct_answers, $array['questions'][$index]['answer']);
    }

    $user_answers = [];
    $qChar = chr(0161);
    $score = 0;
    for ($index = 0; $index < $array_size; $index++) {
        $name = $qChar . ($index + 1);
        array_push($user_answers, $_POST[$name]);
        if (strcmp($correct_answers[$index], $user_answers[$index]) == 0) {
            $score++;
        }
    }
    echo("<script>
        window.alert('You have scored $score out of $array_size!');
        window.location.href='../index.html';
        </script>");
}else {
```

Figure 3.4 – quiz.php (Check Answers)

Then the score of the user will appear via an alert. When the “OK” button is clicked, the user will be redirected back to the landing page.

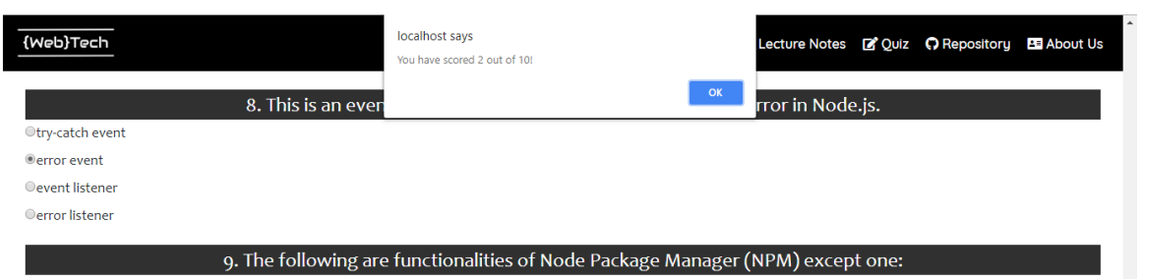


Figure 3.5 – User Score Alert