CSA4378 - Internet Programming

Date: 19/12/24
Day: Thursday

ASSIGNMENT-1

Talkpaka Sathwik
192211910
BE CSE 2022-2026

① SOFTWARE COMPANY SPECIALIZED IN WEB DEVELOPMENT

(i) OS: How do the underlying operating systems (LAMP, WAMP) influence the choice of stack for each project? Discuss the advantages and limitations of each OS in a web development context?

Sol:

- Linux for LAMP:-

* Advantages:- opensource, cost effective, highly customizable, stable. It is widely used in web hosting environments due to it's performance and compatibility with popular web technologies.

* Disadvantages: (i) More Technical Expertise the configure.
(ii) GUI tools are less user-friendly.

- Windows for WAMP:-

* Advantages:- (i) Familiar Interface for users & Administrators.
(ii) Seamless Integration with windows-based tools like Active Directory & MS Office.
(iii) Easier for teams already using windows Environments.

* Limitations:- (i) Higher License costs which stood as drawback.
(ii) Less Prevalent in Hosting Environments Compares to Linux.

(ii) Given the specific needs of Project A and Project B, which stack would you recommend for each Project? Justify your recommendations based on factors like Cost, ease of use, compatibility and scalability?

Sol:

Project-A (Ecommerce platform) → LAMP Stack

• Justification:

* Linux is cost-effective, aligning with the client's limited budget.
* LAMP is highly scalable, making it suitable for a growing business.
* The open-source ecosystem of Linux offers access to a wide range of security tools and community support.

* Project-B -> WAMP Stack
* Justification:

• The client prefers windows, ensuring better compatibility with their existing IT infrastructure.
• WAMP offers straightforward integration with windows applications and services such as MS SQL server with 365 office.

(ii) What types of web applications are typically built using LAMP vs WAMP? Provide examples of real-world applications or platforms that use these stacks?

Sol:

• LAMP:

* Typically used for open-source applications like Word-press, Joomla and Drupal.
* Suitable for web-hosting platforms and frameworks like Laravel.
* Ex: Facebook, Wikipedia and many cloud-based Saas platforms.

• WAMP:

* Used for Internal Business Applications or CMS that require windows Integration.
* Ideal for Enterprise Application leveraging ASP.NET or microsoft technologies.
* Ex: Sharepoint, some intranet portals and windows based CMS systems.

(iv) What Critical factors should developers consider when selecting between LAMP and WAMP for a specific project? Include considerations such as security, comparision on security, community support and ease of Deployment?

Sol:

- Security:-

* LAMP benifits from a robust open-source community for quick patching and updates.
* WAMP offers strong security when Integrated with windows tools like Bit-locker and Active-Directory.

- Performance:-

* LAMP outperforms WAMP in hosting Environments due to the Efficiency of Linux.
* WAMP can be resource heavy but performs well in windows-native environments.

- Community Support:-

* LAMP has the larger community due to it's widespread adoption in hosting and development.
* WAMP's community is smaller, but still active, particularly for enterprise users.

- Ease of Deployment:-

* LAMP requires technical Expertise but is well-documented and flexible.
* WAMP is easier for developers familiar with windows.

(v) Provide Examples of Situations where Choosing one stack over the other might be Crucial? For Instance, how might the choice of stack Impact the long term maintenance and scalability of applications?

* Choosing LAMP:-

* Startups (or) Small bussiness needing cost-effective, scalable

and flexible solutions.

- Example: A Startup building an online marketplace with Dynamic Scalability needs

- Choosing WAMP:

* For organizations with existing Environments in windows technologies.
* Example: An Educational Institution building a CMS Integrated with office 365.

* Question2: Design and develop a fully functional and responsive web form using HTML5 and CSS for a fictional event registration. Your task is to create a user-friendly, accessible and visually appealing form that meets certain Requirements.

* HTML Code with Internal-CSS Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width-device-width,
                                   initial-scale=1.0">
    <title> Event Registration Form</title>
    <style>
        body{
        font-family: Arial, sans-serif;
        background-color: #f4f4f9;
        margin: 0;
        Padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        }
```

```
.form-container{
        background: #ffffff;
        Padding : 20px, 30px;
        border-radius : 10px;
        box-shadow: 0 4px 8px rgba(0,0,0,0.1);
        max-width: 500px;
        width : 100%;
}
h1 {
        Color: #333;
        text-align : center;
        margin-bottom: 20px;
    }
label{
            display : block;
            margin-bottom: 5px;
            color : #555;
        }
input:focus, select:focus, textarea:focus{
            border-color: #007BFF;
            outline : none;
}
.radio-group, .checkbox-group{
                margin-bottom : 15px ;
}
button {
        width : 100%;
        background-color: #007BFF;
        color : #fff;
        Padding : 10px;
        border: none;
        border-radius: 5px;
        font-size : 18px;
        cursor : pointer;
}
```

```
button: hover {
        background-color: #0056b3; }
@media (max-width: 600px)
{
    .form-container { padding: 15px; }
    h1 { font-size: 1.5rem; } }
</style>
</head>
<body>
<div class="form-container">
    <h1> Event Registration </h1>
    <form>
        <label for="full-name"> Full Name </label>
        <input type="text" name="full-name" required>
        <label for="email"> Email Address </label>
        <input type="email" name="email" required>
    <label for="phone"> Phone Number </label>
    <input type="tel" id="phone" name="phone" required>
    <label for="dob"> Date of Birth </label>
    <input type="date" name="dob" required>
    <div class="radio-group">
    <label> Gender </label>
    <label> <input type="radio" name="gender" value="male">
                Male </label>
    <label> <input type="radio" name="gender" value="female">
                Female </label>
    </div>
    <div class="check-box group"
```

```
<label> Hobbies/Interest </label>
<label><input type="checkbox" name="hobbies" value="reading">
Reading </label>
<label><input type="checkbox" name="hobbies" value=
"travelling">Travelling
</label>
<label><input type="checkbox" name="hobbies" value
="gaming">Gaming
</label>
</div>
<label for="event-type"> Event type </label>
<select id="event-type" name="event-type" required>
    <option value=""> Select an Event </option>
    <option value="workshop"> Workshop </option>
    <option value="Seminar"> Seminar </option>
    <option value="Conference"> Conference </option>
</select>
<label for="comments"> Additional Comment </label>
<textarea id="Comments" name="comments" rows="4">
</textarea>
<button type="submit"> Register </button>
</form>
</div>
</body>
</html>
```

① Design choices:

* HTML5 Elements:
* used <input> with specific types like email, tel and date for better data validation and user Experience.

* <textarea> and <select> ensure flexibility for user input.
* Semantic elements like <label> improve accessibility and usability.
* CSS Styles:-
  - A clean, modern look with subtle shadows and rounded corners enhances visual appeal.
  - Hover and focus effects provide immediate feedback during user interaction.

② User-friendly and Accessible Design:-

* Accessibility:- Ensured proper <label> with screen readers.
* Focusses styles for inputs aid keyboard navigation.
* Responsiveness:- Used relative units and media queries to adapt the form for all screen sizes.
* Error-prevention:-
  - Used HTML5 validation for required fields and specific input types.

③ Responsive Design Considerations:-

* Adapted the layout to shrink and fit mobile screens without horizontal scrolling.
* Challenges:- (i) Maintaining spacing and proportions.
  (ii) which were addressed with flexible padding and font sizes.