

- ❖ AUTHOR'S NAME: - **KASHISH PARESH GADA.**
- ❖ MATRICULATION NUMBER: - **102202657.**
- ❖ TUTOR NAME: - **THOMAS KOPSCH.**
- ❖ DATE: - **18.01.2024.**
- ❖ COURSE NAME: - **WEB APPLICATION DEVELOPMENT_ DLBCSWAD01.**
- ❖ COURSE OF STUDY: - **BACHELOR OF COMPUTER SCIENCE.**
- ❖ TITLE OF THESIS:- **ADVANCED WORKBOOK.**

TABLE OF CONTENTS:

Task 1-	2
Task 2-	4
Task 3-	5
Task 4-	6
Task 5-	7
Task 6-	8

Task 1-

Your company asks you to perform an analysis in order to select a new development environment for website projects. This development environment should support HTML, CSS, and JavaScript and have various features to support the development process. Select any three development environments that meet the above requirements and compare them to each other based on a set of criteria defined by you (these criteria should cover at least ten important functionalities for virtual projects). Then, draw a final conclusion: which of these solutions would you recommend to your company, and why?

A1- After evaluating various development environments for website projects that support HTML, CSS, and JavaScript, I have selected Visual Studio Code (VS Code), Sublime Text, WebStorm as the three options and we can compare them on the major criteria:

- Code Editing Features:
 - VS Code- IntelliSense, good features, syntax highlighting.
 - Sublime Text- Customizable editor, auto completion, lightweight.
 - WebStorm- Advanced editing, good debugging tool, code completion feature.
- Collaboration and Remote Development:
 - VS Code- Collaborative editing feature, debugging.
 - Sublime Text- Remote file editing, limited collaborative editing.
 - WebStorm- Supports remote development and deployment.
- Customizability and Themes:
 - VS Code- Filled with different themes and customizable options.
 - Sublime Text- Offers customization for themes and layouts.
 - WebStorm- Least customization option available.
- Version Control Integration:
 - VS Code- Built in source control feature, git integration.
 - Sublime Text- Requires Git integration, dependable.
 - WebStorm- Native Git integration, conflict resolution.
- Integrated Package Manager:
 - VS Code- User-friendly interface.
 - Sublime Text- Requires external package managers.
 - WebStorm- Integrated npm support with GUI.

- Extensions and Plugins:
 - VS Code- Big extension library for customization.
 - Sublime Text- Wide plugin range for workflows and different language.
 - WebStorm- Focused plugins for web development projects.
- Intelligent Refactoring and Code Navigation:
 - VS Code- Intelligent navigation with refactoring support.
 - Sublime Text- Basic navigation with limited refactoring support.
 - WebStorm- Advanced navigation with excellent refactoring support.
- Live Preview and Debugging:
 - VS Code- Live preview through extensions.
 - Sublime Text- Requires external plugins and tools.
 - WebStorm- Built-in live preview, powerful debugging tools.
- Task Automation:
 - VS Code- Built-in system support, integrated terminal for tasks.
 - Sublime Text- Required external task automation plugins.
 - WebStorm- Built-in task automation with grunt or gulp.
- Accessibility Tool:
 - VS Code- Accessibility testing extensions included.
 - Sublime Text- Requires accessibility plugins.
 - WebStorm- Built-in accessibility inspections

Visual Studio Code (VS Code) emerges as the preferred development environment for web projects. We can get one of the best editing features, extension support, debugging capabilities, intelligent code refactoring and navigational tools. In today's vast marketplace and all diverse types of project requirements VS Code has the capabilities to personalize their environment according to the developers.

Task 2-

Your website has become outdated. You need to develop a new, more responsive website that better represents your company. Instead of developing the style sheets for this website from the ground up, you have decided to employ a CSS framework. It must be an alternative to Bootstrap. Which CSS framework would you recommend, what are the limitations of the selected framework, and what should be considered when developing responsive websites?

A2- For a responsive web development project as a substitute for Bootstrap, I suggest opting for Tailwind CSS. Tailwind CSS adopts a utility-first methodology, empowering developers to construct efficient and extensively personalized designs without relying on predefined components.

Nevertheless, it is crucial to acknowledge certain constraints. Tailwind comes with a potentially steeper learning curve compared to Bootstrap, requiring familiarity with its utility classes. Additionally, the file size might be larger due to the extensive utility classes, impacting page load times. While Tailwind offers substantial customization options, it may necessitate advanced CSS knowledge for intricate design adjustments.

In the process of developing responsive websites, several considerations come into play:

1. Responsive Grid System: Employ the framework's grid system for adaptable layouts across diverse devices.
2. Fluid Images and Videos: Assure that media elements adjust seamlessly to varying screen sizes.
3. Mobile-First Approach: Emphasize designing for smaller screens and gradually enhance for larger ones.
4. Media Queries: Deploy media queries for styling modifications at distinct breakpoints.
5. Touch and Mobile Interactions: Craft touch-friendly elements and interactions to optimize the mobile user experience.

In summary, Tailwind CSS stands out as a recommended alternative due to its adaptability, flexibility, and customization features. However, developers need to be aware of the learning curve and potential file size implications. Adhering to responsive design best practices, such as prioritizing a mobile-first approach and leveraging responsive grid systems, remains pivotal for achieving a well-optimized and visually appealing website.

Task 3-

You work in a consulting firm and are expected to advise one of your clients on the development of their new, secure website. How do you proceed? What recommendations do you give to your client? Finally, where should the client focus their attention—first, during the development and, later, when operating the completed website?

A3- In advising our client on the development of their new, secure website, I would adopt a strategic approach encompassing several key recommendations.

❖ During Development:

1. Security-First Approach: Stress the importance of integrating security measures right from the project's initiation to establish a robust defence against potential threats.
2. Authentication and Authorization: Implement robust user authentication methods and detailed authorization controls to protect sensitive information and prevent unauthorized access.
3. Regular Security Audits: Conduct thorough security audits throughout the development process, identifying and addressing vulnerabilities promptly.
4. Content Security Policy (CSP): Employ CSP headers as a preventive measure against risks related to cross-site scripting (XSS) and other potential code injection attacks.

❖ During Operation:

1. Regular Updates and Patching: Uphold the website's security by promptly applying software updates and patches, including those for the underlying server, content management system, and any third-party libraries.
2. Continuous Monitoring: Implement continuous security monitoring to detect and respond to potential threats in real-time, guaranteeing the ongoing protection of the website.
3. Backup and Recovery Plans: Establish robust backup and recovery procedures to mitigate the impact of potential security incidents or data loss, ensuring business continuity.
4. Incident Response Plan: Develop a comprehensive incident response plan outlining steps to take in the event of a security breach, facilitating a swift and effective response.

By focusing on these recommendations during both development and operation phases, the client can not only create a secure website but also sustain its protection against emerging cyber threats. Regular assessments, user education initiatives, and proactive monitoring are critical elements in maintaining a resilient and secure online presence.

Task 4-

JavaScript can be used for interactive web pages and to develop algorithms that solve mathematical problems, for example, by outputting the Fibonacci sequence with arbitrary start and end points. Choose a more complex mathematical problem and create JavaScript code that represents it. Examples of suitably complex problems include approximation of polynomials, Fourier analysis, statistical methods (e.g., linear regression), mortgage repayments, and univariate optimization methods. The output does not have to be generated via a web page: it can be displayed via the console. The source code should be commented line by line.

A4- Let us consider an example of implementing the Newton-Raphson method for finding roots of a mathematical function.

CODE:-

```
// Newton-Raphson method to find the square root of a number
unction squareRootApproximation(number, iterations) {
  // Initial guess (you can choose any reasonable starting point)
  let guess = number / 2;

  // Iterate to refine the approximation
  for (let i = 0; i < iterations; i++) {
    // Update the guess using the Newton-Raphson formula
    guess = 0.5 * (guess + number / guess);
  }
  return guess; // Final approximation of the square root
}
```

EXAMPLE:-

```
// Example usage
const inputNumber = 25; // Replace with the desired number
const numberOfIterations = 5; // Choose the number of iterations

// Calculate and display the square root approximation
const result = squareRootApproximation(inputNumber, numberOfIterations);
console.log(`Square root approximation of ${inputNumber} after
${numberOfIterations} iterations: ${result}`);
```

Task 5- Choose a common trend in internet development (e.g., serverless architecture) and describe its impact on web application development, both today and in the future.

A5- A significant trend in internet development is the increase in Serverless Computing. This architectural approach empowers developers to construct and operate applications without the need to oversee the underlying server infrastructure. Instead of dealing with server provisioning and management, developers can concentrate on code creation, allowing cloud providers to manage execution, scaling, and maintenance.

- **Current Impact:** Serverless architecture simplifies development workflows, accelerates deployment, and reduces operational overhead. It operates on a pay-as-you-go model, where users only pay for the actual compute resources consumed during execution. This results in cost efficiency and resource optimization. Additionally, serverless platforms like AWS Lambda, Azure Functions, and Google Cloud Functions support a wide range of programming languages, enhancing flexibility for developers.
- **Future Implications:** The future trajectory of web application development is poised to witness a heightened embrace of serverless computing, driven by its scalability, cost-effectiveness, and user-friendly nature. Developers can anticipate broader language support, improved cold start times, and enhanced tooling for debugging and monitoring within serverless platforms. As serverless architectures progress, they may expand beyond function execution to encompass comprehensive application workflows, establishing a versatile paradigm for diverse use cases.

Serverless computing fits well with the industry's shift towards microservices and containerization, providing a more detailed and efficient way to use resources. The future integration of serverless technologies is expected to reshape how applications are developed, deployed, and managed, emphasizing a more agile and cost-effective paradigm for web development.

Task 6- Your company tasks you with developing an order form for computer components.

Develop a web page using HTML, CSS, and JavaScript to demonstrate how the solution might function and appear visually. Please develop the web page as a “one pager.” External links to available frameworks are allowed.

A6-

CODE:-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Computer Component Order Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }
    header {
      background-color: #333;
      color: #fff;
      text-align: center;
      padding: 2em;
    }
    section {
      margin: 3em;
    }
  </style>
</head>
<body>
  <header>
    <h1>Computer Component Order Form</h1>
  </header>
  <section>
    <h2>Order Form</h2>
  </section>
</body>
</html>
```



```

label {
    display: block;
    margin-bottom: 0.5em;
}

input, select {
    width: 100%;
    padding: 0.5em;
    margin-bottom: 2em;
    box-sizing: border-box;
}

button {
    background-color: #4caf50;
    color: white;
    padding: 0.7em;
    border: none;
    cursor: pointer;
    width: 100%;
}
</style>
</head>
<body>
    <header>
        <h1>Computer Components Order Form</h1>
    </header>
    <section>
        <label for="component">Select Components:</label>
        <select id="component">
            <option value="Keyboard">Keyboard</option>
            <option value="Mouse">Mouse</option>
            <option value="Printer">Printer</option>
            <option value="storage">Storage</option>
        </select>
    </section>

```

```

        <label for="quantity">Quantity:</label>

        <input type="number" id="quantity" min="1"
placeholder="Enter quantity">

        <button onclick="submitOrder()">Submit Order</button>

    </section>

<script>
    function submitOrder() {
const component = document.getElementById('component').value;
const quantity = document.getElementById('quantity').value;

    alert(` Order Placed!\nComponent: ${component}\nQuantity:
${quantity}`);

    }
    </script>
</body>
</html>

```

OUTPUT:-

Computer Components Order Form:-

Computer Components Order Form

Select Components:

Keyboard

Quantity:

Enter quantity

Submit Order

Components Dropdown Box:-

Computer Components Order Form

Select Components:

Keyboard

Keyboard

Mouse

Printer

Storage

Enter Quantity

Submit Order

Order Placed Alert:-

This page says

Order Placed!
Component: Printer
Quantity: 2

OK

Select Components:

Printer

Quantity:

2

Submit Order