KennethHefford_T1A1 Workbook

Questions

Q1 Identify and explain common and important components and concepts of web development markup languages

HyperText Markup language or commonly known as HTML is a markup language for creating documents for web development to be viewed in a web browser. It is unlike a programming language as markup langaugaes are static, they do not include logic. They define style, structure and logic.

HTML documents comprise of HTML elements. It is a component of an HTML document that provides structure and interprets that part of the HTML document.

A HTML Element is made up of a starting tag and a closing tag. Such as for paragraphs on a webpage; `<p> some text </p>`.

A HTML attribute provides additional information about elements. e.g. `<img src="dog.jpg">` `<img>` being the tag and src shows where the path to the image is.

Markup languages have many applications including webpages, word processing and transfer of information between systems such as XML.

Q2 Define the features of the following technologies that are essential in terms of the development of the internet:

- packets
- IP addresses (IPv4 and IPv6)
- routers and routing
- domains and DNS

Explain how each technology has contributed to the development of the internet.

1. Packets:
    - in networking, a packet is a small segment of data, but part of a larger piece of data. i.e. data sent from the internet is divided into packets, which contains a portion of the larger data, sent separately.These are handled by routers and switches to get to their destination.
    - this allowed for robust transfer of information as if a packet was lost, only that data needed to be retransmitted. Can be done on a large scale and efficient packets allow for many devices to share network resources simultaneously.
2. IP addresses (IPv4 and IPv6):
    - Internet protocol, defines the method by which data is sent from one computer to another on the internet. IP addresses allowed devices to locate each other on the internet.

        - IPv4: 32 bit addresses in decimcal notation e.g. 192.168.3. Limited addresses (4.3 bilion)
        - IPv6: 128 bit addresses in hexadecimal 2001:0db8:85a3:0000..... 340 undecilion unique addresses. This allowed for improved security and better support for mobile devices.

3. Routers & Routing:
    - Routers are devices that connect different devices, networks and direct data traffic across the internet. They determine the best path for data packets using tables.
    - Routing is the process of determining the optimal path for data packets. Ensures data is transmitted efficiently with a low tolerance for packet loss.
    - routing allows for interconnection of networks and scalability. Routing ensure reliable and timely delivery of packets across complex networks.
4. Domains and DNS (Domain Name System):
    - Domains are readable labels representing locations or resources on the internet like www.website.com. Can have .com, .org, and country labels like .com.au.
    - DNS turns domains into IP addresses. Allows users to access websites and services using domain names.

In conclusion, IP addresses enable global communication, packets optimise data transmission, routers interconnect networks, and DNS ensures seamless navigation by translating domain names into IP addresses. Together, these technologies form the backbone of the interconnected world we know as the internet.

Q3 Define the features of the following technologies that are essential in terms of the development of the internet:

- TCP
- HTTP and HTTPS
- web browsers (requests, rendering and developer tools)

Explain how each technology has contributed to the development of client and server communication over the internet (50 - 150 words for each technology)

1. TCP (Transmission Control Protocol): Features: Reliable Communication: TCP ensures reliable data transmission between devices over IP networks. It guarantees that each bit sent from the source host reaches the destination host. Three-Way Handshake: TCP establishes a connection-oriented communication model through a three-way handshake process. This ensures a reliable session between the client and server. Sequencing and Acknowledgment: TCP assigns sequence numbers to data packets, ensuring correct order and minimising data loss. Acknowledgments confirm successful receipt. Flow Control: TCP uses sliding window protocol to manage data flow, preventing congestion and optimising transmission. Contribution to Client-Server Communication: TCP's reliability and connection-oriented model form the foundation for most internet-based applications. It enables web browsing, email, file transfers, and more by ensuring data integrity and order.
2. HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure): HTTP Features: Client-Server Communication: HTTP allows web clients (browsers) to request resources (HTML, images, etc.) from web servers. Statelessness: Each request is independent, without knowledge of previous requests. Cookies make it stateful for user authentication. HTTP Methods: GET (retrieve data), POST (send data),

PUT (update data), DELETE (remove data). HTTPS Features: Security: HTTPS enhances communication security by encrypting data using SSL/TLS protocols. Authentication: SSL certificates validate server identity, ensuring users connect to legitimate websites. Data Integrity: HTTPS prevents eavesdropping and tampering during transmission. Contribution to Client-Server Communication: HTTP forms the basis for web communication, while HTTPS adds encryption and security. Understanding these protocols is crucial for network engineers and cloud professionals

3. Web Browsers: Request and Rendering: Request: Browsers send HTTP requests to web servers when users access websites. These requests fetch HTML, CSS, images, and other resources. Rendering: Browsers interpret HTML, CSS, and JavaScript to render web pages for users. Developer Tools: Debugging: Browsers provide developer tools (e.g., console, viewport sizes) for testing, analysing, and optimising web pages. CSS Grid Tool: Browsers like Firefox offer tools for custom CSS grids. Performance and Memory Tools: Browsers aid in web optimisation and debugging memory leaks.

Q4 Describe the features of interpreters and compilers and how they are different.

Interpreters:

- Line-by-Line Execution: Interpreters read and execute the source code line by line.
- Immediate Feedback: Errors are detected during execution, allowing quick debugging.
- No Intermediate Code: Interpreters do not generate intermediate machine code.
- Slower Execution: Interpreted code runs slower because each line is analysed during runtime.
- Examples: Python, JavaScript.
- Interpreters are useful for scripting languages and dynamic web applications. They allow rapid development and easy debugging but sacrifice execution speed.

Compilers:

- Full Program Compilation: Compilers analyse the entire program before execution. Optimised Code Generation: Compilers generate efficient machine code.
- Object Code: Compilers produce object files that need linking to create an executable.
- Faster Execution: Compiled code runs faster due to pre-optimisation.
- Examples: C, C++, Java.
- Compilers are essential for performance-critical applications. They ensure efficient execution but require longer development cycles.

Q5 Identify TWO commonly used programming languages and explain the benefits and drawbacks of each.

1. Python:

Advantages:

- Easy to Learn: Python has a simple and readable syntax, making it ideal for beginners.
- Versatility: Python is used in web development, data analysis, artificial intelligence, machine learning, and automation.
- Extensive Library Support: Python's rich ecosystem includes libraries like NumPy, pandas, and TensorFlow.

Disadvantages:

- Slower Execution Speed: Python is an interpreted language, which can lead to slower performance compared to compiled languages.
- Less Optimal for Mobile App Development: Python is not the best choice for resource-intensive mobile apps.

2. JavaScript:

Advantages:

- Runs Natively in Browsers: JavaScript powers web interactivity and runs directly in browsers.
- Large Ecosystem: JavaScript has a vast ecosystem of libraries (e.g., React, Vue.js) for web development.
- Asynchronous Programming: JavaScript supports asynchronous operations, crucial for responsive web applications.

Disadvantages:

- Inconsistent Browser Support: Different browsers may interpret JavaScript differently.
- Limited Standard Library: JavaScript lacks a comprehensive standard library compared to other languages.

Q6 A hypothetical client has sent you an email (shown in the Q6 Email section), asking for you to build them a website. Write an appropriate, professional email response that shows your understanding of the client's needs for the website, as well as an understanding of appropriate technologies or tools needed to build the website yourself.

Dear Alex,

Thankyou so much for your email. I appreciate the opportunity to collaborate on your museum's website. Let's create an engaging online presence that aligns with your vision.

Website Goals:

1. Showcase Exhibits: We'll feature captivating exhibits with high-quality images.
2. Visitor Information: Essential details like hours, location, and admission fees.
3. Contact Options: Easy ways for visitors to reach out.

Proposed Approach:

Responsive Design: Adapt seamlessly to various devices. Use HTML, CSS, and Bootstrap.

Homepage: Engaging hero section. Briefly introduce the museum's mission.

Exhibits Page: Organized gallery with descriptions. Interactive elements (virtual tours).

Visit Us Page: Address, hours, directions. "Plan Your Visit" CTA.

Contact Page: Contact form, phone numbers, email.

About Page: Museum history and team.

SEO: Optimize for search engines.

User-Friendly Navigation: Clear menus.

Visual Elements: High-quality multimedia.

Feel free to share any specific preferences or ideas you have, and we'll bring your vision to life. Looking forward to collaborating with you!

Best regards, Kenneth Hefford Senior Web Developer.

Q7 Think back to a scenario or situation in your own software development projects or work. Explain how you would do things differently if you had a chance to go through that scenario again, using an appropriate reflective cycle or reflection technique.

I would implement some agile methodologies.

In my portfolio website. There were often times I would be going beyond the scope of my abilities and the project. During the planning phase I tried to ensure I added all features to a website that should have been simpler. Some features were uneccessary and overkill. Due to this the project took longer.

Planning

1. I would focuse on conceptualising and planning better to understand the goals and requirements and the contraints in the time frame given. Prioritise features that are essential.

Iteration

2. Develop the code in shorter iterations as trying to cram it all in one large block made me prone to bugs and mistakes. Continuously test whether the website was working well and ask for feedback from friends.

Release & Deployment

3. Test the final product. I left this quite late and realised the website did not function correctly in Safari but worked in Chrome. Due to time constraints website was deployed with this issue.

Continuous Improvement

4. regularly assess what was going well and what needed improvement. Adjust my processes to make things more efficient and streamline. Redo iteration with these new reflections in mind.

Q8 A large part of career growth as an information technology professional happens through networking and workshops, often found at online or in-person events or workshops. Create an action plan that identifies several relevant networking opportunities for you to participate in or attend, and add some information about what you expect to gain or grow through each item in the action plan.

1. Online IT Workshops and Webinars:

- Expectations:
    - Gain insights into emerging technologies (e.g., AI, cloud computing).
    - Learn practical skills (coding, cybersecurity, data analytics).
    - Connect with industry experts and fellow learners.
- Platforms:
    - Coursera, university, udemy, and LinkedIn learning.

2. Local Tech Meetups and Conferences:

- Expectations:
    - Expand your network by meeting local professionals.
    - Discuss trends, challenges, and best practices.
    - Discover job openings or freelance opportunities.
- Platforms:
    - Meetup.com, Eventbrite, linkedin.

3. Industry-Specific Webinars:

- Expectations:
    - Deepen domain knowledge (e.g., medical tech, fintech).
    - Connect with peers in your niche.
    - Stay updated on industry regulations and innovations.
- Platforms:
    - Look for webinars hosted by industry associations or specialized platforms.

4. LinkedIn and Professional Networking Sites:

- Expectations:
    - Build a strong online presence.
    - Connect with colleagues, mentors, and recruiters.
    - Connect with employers actively hiring.
- Platforms:
    - LinkedIn

5. Attend Local Hackathons or Coding Challenges:

- Expectations:
    - Collaborate with other developers.
    - Solve real-world problems.

    - Showcase your coding skills.
- Platforms: University events.

Q9 Explain the uses of language-learning model technologies (such as ChatGPT) on written and technical works, such as reports and software projects.

Writing Assistance:

LLMs provide feedback on essays and reports, helping students improve their writing. They suggest improvements in grammar, coherence, and clarity.

Code Generation and Documentation:

LLMs assist in writing code snippets and comments. They can generate descriptive code explanations and API documentation.

Technical Reports and Summaries:

LLMs help create clear technical reports and summaries. They simplify complex concepts for readers.

Natural Language Interfaces:

LLMs enable conversational interactions with software. Users can query databases or control applications using natural language.

Sentiment Analysis and Feedback:

LLMs analyze user feedback, aiding product improvement.

Content Summarization:

LLMs summarize lengthy documents, making information more accessible.

Translation and Multilingual Support:

LLMs assist in translating software content for global audiences.

Natural Language Search: LLMs enhance search engines by understanding user queries.

Q10 Explain the legal and ethical impacts of the usage of language-learning model technologies (such as ChatGPT) in written and technical works, such as reports and software projects.

1. Legal Impacts:

Copyright Concerns: LLMs might create content that violates copyrights. Users should be cautious.

Liability: If LLMs give incorrect legal advice, who is responsible?

Privacy: LLMs could accidentally reveal sensitive information.

2. Ethical Considerations:

Bias: LLMs learn from data, so they inherit biases. Developers must ensure fairness.

Transparency: Users should know when they're interacting with an AI.

Misinformation: LLMs can generate false content unintentionally.

Human Oversight: Balance automation with human judgment.

Software Development: LLMs can generate code, but ethical use is essential. Consider licensing and intellectual property rights.

User Consent and Privacy: LLMs process user data; consent and privacy matters.

Q11 Explain multiple skills from each of the categories below, and how they're useful to a software development workplace.

1. Soft skills:

Communication Skills:

Effective communication is crucial for collaborating with team members, understanding requirements, and explaining technical concepts to non-technical stakeholders. Clear verbal and written communication, active listening, and empathy foster better collaboration and understanding.

Problem-Solving and Critical Thinking:

Software development involves solving complex problems. Being able to analyze issues, identify patterns, and devise creative solutions is essential. Break down problems, consider alternative approaches, and think logically to find optimal solutions.

Teamwork and Collaboration:

Software projects are rarely done alone. Working well with others, respecting diverse viewpoints, and contributing positively are vital. Active participation in team discussions, willingness to share knowledge, and supporting team goals.

Adaptability and Flexibility:

The tech landscape evolves rapidly. Developers must adapt to new tools, languages, and methodologies. Learn continuously, and be open to adopting new practices.

Time Management and Prioritization:

Deadlines are common in software development. Efficiently managing tasks ensures timely delivery. Use tools like calendars, task lists, and agile methodologies to organize work.

2. Hard skills:

Programming Languages:

Proficiency in languages like Python, Java, C++, or JavaScript is fundamental for writing code. learn syntax, data structures, algorithms, and best practices for each language.

Version Control (e.g., Git):

Collaborative development requires version control. Git helps manage code changes, merge branches, and track history. Understand Git commands, branching, and workflows.

Web Development (Frontend and Backend):

Building web applications involves frontend (HTML, CSS, JavaScript) and backend (server-side languages, APIs). Learn frontend frameworks (e.g., React, Angular) and backend technologies (e.g., Node.js, Django).

Testing and Debugging:

Bugs are inevitable. Proficiency in testing (unit, integration, and end-to-end) and debugging is essential. Learn testing frameworks debugging tools like chrome dev tool

Q12 Explain multiple roles or job positions that would be found in a medium-sized software development company.

Product Manager (PO):

- Responsibilities: Oversees the entire development process, sets strategic goals, and monitors product Key Performance Indicators (KPIs).
- Collaboration: Works closely with developers, product owners, marketing specialists, sales representatives, and stakeholders.

Business Analyst (BA):

- Responsibilities: Gathers and analyzes information about the product, acts as a bridge between market needs and engineering, and provides recommendations for product improvement.
- Focus: Studies user behavior, concerns, and evaluates the product's impact on solving market problems.

Engineering Manager:

- Responsibilities: Manages team dynamics, establishes collaborative working conditions, and ensures the optimization of the software development team structure.
- Expertise: Possesses a strong technical background, selects optimal engineering solutions, and analyzes potential challenges.

Software Architect:

- Responsibilities: Makes decisions on the internal arrangement of software based on business needs, outlines technical instruments, and designs the system.
- Vision: Plans software enhancement, extension, and addition of new features with a strategic perspective.

Software Developers:

- Responsibilities: Code the software using various programming languages, frameworks, and libraries, with different levels of expertise (junior, middle, senior).
- Collaboration: Work closely with designers, testers, product engineers, and other core team members to ensure parallel progress.

Bibliography

Sadok, S.B. (2023) What is markup language? examples, types & definition, Semrush Blog. Available at: https://www.semrush.com/blog/markup-language/ (Accessed: 21 April 2024).

Cloud Flare (no date) What is the internet protocol? | cloudflare, cloudflare.com. Available at: https://www.cloudflare.com/en-gb/learning/network-layer/internet-protocol/ (Accessed: 21 April 2024).

Cristian (2023) Demystifying TCP: Understanding the fundamentals of reliable network communication, CloudPunk. Available at: https://www.cloudpunk.blog/post/demystifying-tcp-understanding-the-fundamentals-of-reliable-network-communication (Accessed: 21 April 2024).

Ghate, S. (2020) HTTP made easy: Understanding the web client-server communication, HackerNoon. Available at: https://hackernoon.com/http-made-easy-understanding-the-web-client-server-communication-yz783vg3 (Accessed: 21 April 2024).

GeeksforGeeks (2023) Browsers role in web development, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/browsers-role-in-web-development/ (Accessed: 21 April 2024).

Interpreter vs compiler: Differences between interpreter and compiler (no date) Programiz. Available at: https://www.programiz.com/article/difference-compiler-interpreter (Accessed: 21 April 2024).

Fisher, R. (2023) The 10 most popular programming languages: At a glance, Medium. Available at: https://theskillseeker.medium.com/the-10-most-popular-programming-languages-at-a-glance-a1781dc0e4cb (Accessed: 21 April 2024).

Team, S. (2024) Large language models (llms): Technology, use cases, and challenges, Swimm. Available at: https://swimm.io/learn/large-language-models/large-language-models-llms-technology-use-cases-and-challenges (Accessed: 21 April 2024).

Trancoso, I. et al. (1970) The impact of language technologies in the legal domain, SpringerLink. Available at: https://link.springer.com/chapter/10.1007/978-3-031-41264-6_2 (Accessed: 21 April 2024).

(No date) Software company roles | indeed.com. Available at: https://www.indeed.com/career-advice/career-development/software-company-roles (Accessed: 21 April 2024).