1. **Describe how HTML fits into the broader ecosystem of a website. Contrast the fundamental role of HTML with the primary roles of CSS and JavaScript. (5 points)**
   1. HTML creates the structure and content of a website — the headings, paragraphs, links, and overall layout that define what is on the page. CSS handles the presentation layer, controlling colors, fonts, and positioning. JavaScript adds interactivity and dynamic behavior, allowing the page to respond to user actions and update without reloading.
2. **Explain the difference between HTML structure and HTML semantics. Why is writing semantic HTML considered a best practice? Provide one example of a semantic HTML element and one example of a non-semantic element. (10 points)**
   1. HTML structure describes how elements are organized and nested to create the layout of a page. HTML semantics describes the meaning of those elements using tags that reflect what the content represents, not just how it looks. Writing semantic HTML is best practice because it improves accessibility for screen readers, helps search engines understand the content, and makes code easier to maintain. An example of a semantic element: <nav>, an example of a non-semantic element: <div>
3. **What is the "three-tier model" (also known as three-tier architecture) in web systems? Briefly describe the function and responsibility of each of the three tiers. (15 points)**
   1. The three-tier model is an architecture that is separated into 3 layers. The presentation tier(client) consists of the front end of the website, including HTML, CSS, and JavaScript. It displays information to users and handles input.. The logic tier(server) consists of the backend layer that processes input, applies business logic, and communicates with databases. The data tier(database) consists of the storage layer that manages and retrieves data such as user accounts, products, or posts
4. **Explain what is meant by a Universal Interface in a REST API. (5 points)**
   1. A Universal Interface means all clients interact with a REST API consistently and predictably. It uses standard HTTP methods, uniform resource identifiers, and self-descriptive messages. This consistency allows clients and servers to evolve independently.
5. **Explain how your browser chooses which CSS rule to apply to a tag in the case where there are multiple rules that could apply. (15 points)**
   1. When there are multiple the browser follows the cascade and specificity hierarchy. First, important rules override normal ones. Then the origin of the styles takes effect: Inline styles(first) → internal stylesheet → external stylesheet → browser default(last). After that comes specificity: inline styles(first) → ID selectors(#id) → class, attribute, pseudo-class → Element and pseudo-element selectors. If there is a tie, the last rule defined in the CSS wins.