Short Answer:

Answer the following questions with complete sentences in **your own words**. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers.

1. **What are static websites?**

Static websites uses server-side rendering to serve pre-built HTML, CSS, and JavaScript files

2. **What are dynamic websites?**

Dynamic websites area websites that display different of content when a user views the site.

3. **What are the major differences between static and dynamic websites?**

-Static HTML have pre-built HTML while Dynamic HTML is generated by request.
-Performance for Statis website is quick while Dynamic website is server processing.
-Scalability for Static website involves creating new files for each page or manually modifying each page while Dynamic websites reuses the same page template.
-Personalization for static HTML can be cumbersome while Dynamic pages are simple

4. **What is first contentful paint and time to interactive?**

First contentful Paint (FCP) is an important, user-centric metric for measuring load speed because it marks the time where the user can see anything on the screen

Time to Interactive(TTI) is a performance metric that measures a page's load responsiveness and helps identify situations where a page looks interactive but actually isn't.

5. **What is SSR, CSR, and SSG? When would you use each one?**

-SSR(Server-side Rendering) generates the HTML for a webpage on the server side.
-CSR(Client-side Rendering) generates the HTML components on the browser side, by executing Javascript code within the browser that manipulates the HTML DOM to build the HTML nodes.
-SSG(Static Side Generator) generates a full static HTML website based on raw data and a set of templates

6. **How do they affect page performance?**

-SSR(Server-side Rendering) generates faster Initial page loads.
-CSR(Client-side Rendering)'s response time is greatly improved because there is no round trip to the server. However, it has to wait for the JavaScript to load first and start processing.
-SSG(Static Side Generator) creates webpages in advance instead of on demand and so webpages load slightly faster in users' browsers.

7. **What are template engines and their advantages?**

Template Engines are softwares that change static template files into HTML by converting variables into template file with actual values and then transforming the template into a HTML file.

Advantages
- It encourages good code organization
- Output is more expressive because it doesn't require many string concatenation
- it offeres better productivity by handing output encoding, iterating, conditionals, etc.
- it requires less code overall (jade in particular has a very terse syntax)

8. **What is DRY?**

DRY(Don't Repeat yourself) is a principle where you avoid repetition of information and code.

9. **What is CRUD?**

CRUD (Create, Read, Update, Delete) refers to the four functions that are considered necessary to implement a persistent storage application

C: post. R: Get. U: Put/Patch D:Delete

10. **What is a seed file for?**

Seed file are scripts where you store dummy data to test out the web application.

Coding Questions:

Use HTML/CSS/JS to solve the following problems. You are highly encouraged to present more than one way to answer the questions. Please follow best practices when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

1. **To-Do App (Express.js, EJS, SCSS)**
   Use EJS & SCSS to create the frontend for yesterday's to-do application. It should be fully-functional and connected (can send requests) to the server. Postman screenshots are not needed.
   - The to-do app frontend should include the following feature requirements (based on the same routes). Refreshing the page should show the most updated to-do list.

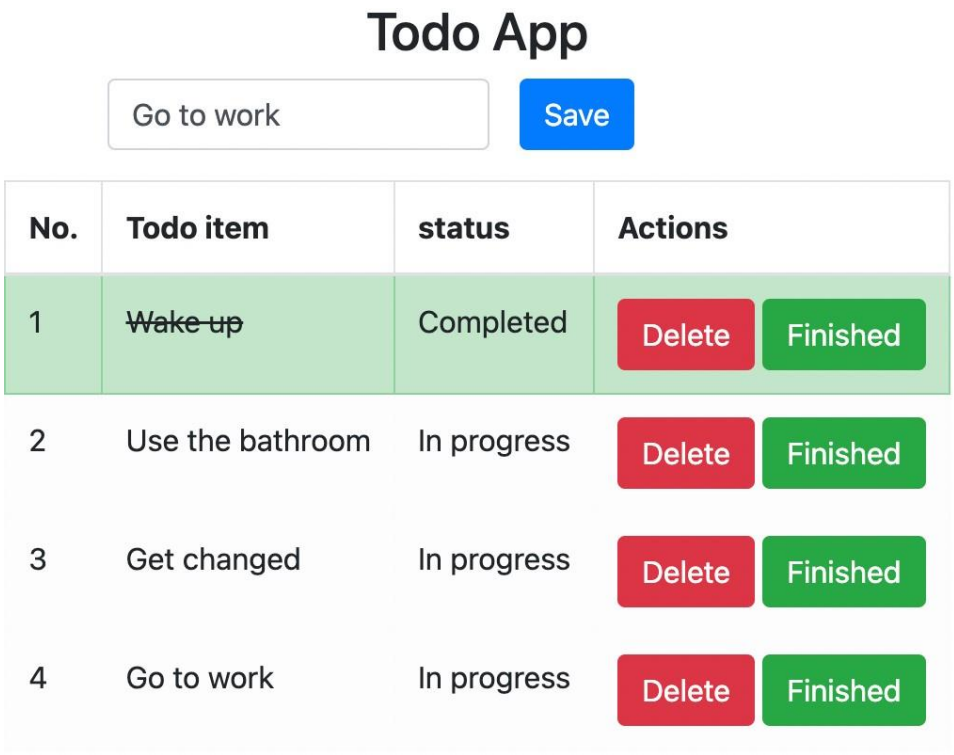
   1. `POST: /todo/{filename}`
      When the user fills out the form to add a new to-do item and clicks "Save", it should send the POST request to the server.
      - Title & description are input text fields, and status & priority are dropdowns.
      - If any of the text fields are empty, it should not send the request, and notify the user appropriately.
      - If there is an error from the server response, display an appropriate error message on the page.
      - If the request was successfully completed, display an appropriate success message on the page.

   2. `GET: /todo/{filename}`
      When the user navigates to this page, the server should render the corresponding EJS template. It should display all the existing to-do items in the file.
      - If there is an error from the server response, display an appropriate error message on the page and redirect to the home page (can be anything).
      - If the request was successfully completed, display an appropriate success message on the page.

3. `PUT: /todo/{filename}`
   Each to-do item should have an “Edit” button that when clicked, allows them to modify the to-do item (which becomes text fields & dropdowns). When the user clicks “Save”, it should send the PUT request to the server.
   - If there is an error from the server response, display an appropriate error message on the page.
   - If the request was successfully completed, display an appropriate success message on the page.

4. `DELETE: /todo/{filename}`
   When the user clicks a “Delete” button to remove this to-do item, it should send the DELETE request to the server.
   - If there is an error from the server response, display an appropriate error message on the page.
   - If the request was successfully completed, display an appropriate success message on the page.

5. Follow the styling below. You should still have the other input fields and display their properties in the table row.



Paired Programming:

Use JS to solve the following problems with your partner. Please remember to label who was acting as the driver and navigator for each problem and write down your feedback on their performance. **Feedback will be confidential**.

Leetcode #409: Longest Palindrome
Leetcode #104: Maximum Binary Tree