WAD viva

1. What is Responsive Web Design?

Responsive Web Design is an approach to web design aimed at creating sites that provide an optimal viewing and interaction experience across a wide range of devices, from desktop computers to mobile phones. It involves using flexible layouts, images, and CSS media queries to adapt the site's design to different screen sizes and orientations.

2. What is Bootstrap?

Bootstrap is a popular front-end framework for building responsive and mobile-first websites and web applications. It provides a collection of CSS and JavaScript components, as well as pre-designed templates and themes, to help developers quickly and easily create modern and responsive user interfaces.

3. What are the benefits (or features) of Bootstrap?

Bootstrap offers several benefits and features including:

- Responsive Grid System: Bootstrap comes with a responsive, mobile-first grid system that adapts to various screen sizes.
- Pre-styled Components: It provides a wide range of pre-styled UI components such as buttons, forms, navigation bars, and more, which can be easily customized and integrated into projects.
- Cross-Browser Compatibility: Bootstrap ensures consistent styling and functionality across different web browsers.
- Extensive Documentation: Bootstrap provides comprehensive documentation and examples to assist developers in using its features effectively.
- Active Community: It has a large and active community of developers, providing support, resources, and third-party plugins/extensions.
- Customizable: Bootstrap allows developers to customize and extend its components to fit their specific project requirements.
- Integration with JavaScript Plugins: It seamlessly integrates with popular JavaScript plugins, enhancing the functionality of web applications.

4. What is CDN?

CDN stands for Content Delivery Network. It is a network of distributed servers that deliver web content to users based on their geographic location. CDNs are designed to improve the performance and reliability of web content delivery by reducing latency and bandwidth usage. When a user requests a piece of content, such as an image or a script, the CDN serves it from the nearest server in its network, rather than the origin server, which can be located far away. This helps to speed up the delivery of content and improve the overall user experience.

—--------------------------

1. What is GitHub?

GitHub is a web-based platform used for version control and collaboration on software projects. It utilizes the Git version control system and provides hosting for Git repositories. GitHub allows developers to store, manage, and share code, as well as track changes made to code over time.

2. What is the use of GitHub?

The primary use of GitHub is to facilitate collaboration among developers working on software projects. It allows developers to:

- Host and share code repositories.
- Collaborate with team members by managing and merging code changes.
- Track issues and bugs, and coordinate work on fixing them.
- Review and discuss code changes using pull requests.

- Showcase projects and contribute to open-source projects.

3. What is the use of `git init`, `git status`, `git add .`, `git commit`, `git push` commands?

- `git init`: Initializes a new Git repository in the current directory.
- `git status`: Displays the current status of the repository, including any changes to tracked files.
- `git add .`: Adds all changes in the working directory to the staging area, preparing them to be committed.
- `git commit`: Records the changes staged in the staging area and creates a new commit in the repository with a commit message describing the changes.
- `git push`: Uploads local repository content to a remote repository, typically on a platform like GitHub, GitLab, or Bitbucket.

4.Explain the steps to push your code to GitHub from Visual Studio Code terminal?

- Open Visual Studio Code and navigate to the project directory.
- Initialize a new Git repository if one hasn't been initialized already using `git init`.
- Add your files to the staging area using `git add .`.
- Check the status of your files with `git status` to ensure all changes are staged.
- Commit your changes using `git commit -m "Your commit message"`.
- Link your local repository to a remote repository on GitHub using `git remote add origin <remote_repository_URL>`.
- Push your changes to the remote repository on GitHub using `git push -u origin master` (or the branch name you are working on). You may need to authenticate with your GitHub credentials.
- Verify that your code has been successfully pushed to GitHub by visiting your repository on the GitHub website.

—---------------------------------

1. What is Docker?

Docker is a platform for developing, shipping, and running applications inside containers. Containers allow developers to package an application with all of its dependencies into a standardized unit for software development, ensuring that it will run consistently across different environments.

2. What is the use of Docker?

The primary use of Docker is to streamline the software development lifecycle by providing a consistent environment for developing, testing, and deploying applications. Docker facilitates the following:

- Isolation: Containers provide a lightweight and isolated environment for running applications, ensuring that dependencies do not interfere with each other.
- Portability: Docker containers can run on any platform that supports Docker, enabling developers to build and deploy applications across different environments with ease.
- Efficiency: Docker containers share the host operating system kernel, making them lightweight and efficient compared to traditional virtual machines.
- Scalability: Docker enables easy scaling of applications by allowing developers to spin up multiple instances of containers to handle increased workload.

3.What is a Docker container?

A Docker container is a lightweight, standalone, and executable package that includes everything needed to run a piece of software, including the code, runtime, system tools, libraries, and settings. Containers are isolated from each other and from the host system, ensuring that applications run consistently regardless of the environment.

4. What do you mean by Docker images?

Docker images are read-only templates used to create Docker containers. An image contains the filesystem and configuration needed to run a container, including the application code, runtime, dependencies, environment variables, and other settings. Images are created using a Dockerfile, which specifies the instructions for building the image.

5. Which commands have you used to implement this assignment?

For implementing Docker-related tasks, common commands include:
- `docker build`: Builds a Docker image from a Dockerfile.
- `docker run`: Runs a Docker container from an image.
- `docker ps`: Lists running containers.
- `docker stop`: Stops a running container.
- `docker push`: Pushes a Docker image to a Docker registry, such as Docker Hub.
- `docker pull`: Pulls a Docker image from a Docker registry.

6. Explain the steps to create Docker images.

The steps to create a Docker image typically involve:
- Writing a Dockerfile that specifies the instructions for building the image, including the base image, dependencies, environment variables, and commands to run.
- Running the `docker build` command to build the image from the Dockerfile.
- Tagging the image with a descriptive name using the `-t` option.
- Optionally, pushing the image to a Docker registry using the `docker push` command to make it available to others.

—--------------------------------------

1. What is AngularJS?

AngularJS is a JavaScript-based open-source front-end web application framework maintained by Google. It is primarily used for developing single-page applications (SPAs) and dynamic web applications. AngularJS extends HTML with additional attributes and provides a framework for building client-side web applications using a Model-View-Controller (MVC) or Model-View-ViewModel (MVVM) architecture.

2. What are the features of AngularJS?

AngularJS offers several features that make it popular among developers:
- Two-way Data Binding: AngularJS automatically synchronizes the data between the model and the view, eliminating the need for manual DOM manipulation.
- Directives: AngularJS provides a set of built-in directives that extend HTML with additional functionality, allowing developers to create reusable components and manipulate the DOM easily.
- Dependency Injection: AngularJS has a built-in dependency injection mechanism that helps manage dependencies and makes components more testable and modular.
- Templating: AngularJS allows developers to create dynamic views using templates, which are HTML files with additional markup and AngularJS expressions.
- MVC Architecture: AngularJS follows the Model-View-Controller (MVC) architecture, which helps organize code and separates concerns, making it easier to maintain and scale applications.
- Services: AngularJS provides a set of built-in services that can be injected into components to perform common tasks such as HTTP requests, localization, and logging.
- Testing: AngularJS has built-in support for unit testing and end-to-end testing, making it easier to write and maintain tests for AngularJS applications.

3. Which commands have you used to implement this assignment?

The commands commonly used to work with AngularJS projects include:
- `ng new <project-name>`: Creates a new AngularJS project.
- `ng generate <schematic>`: Generates components, services, modules, etc., using schematics provided by AngularCLI.
- `ng serve`: Builds and serves the AngularJS application locally for development.
- `ng build`: Builds the AngularJS application for production deployment.
- `ng test`: Runs unit tests for the AngularJS application.

- `ng e2e`: Runs end-to-end tests for the AngularJS application.
- `ng add`: Adds support for additional features or libraries to the AngularJS project.
- `ng update`: Updates dependencies and configurations of the AngularJS project to the latest versions.
- `ng lint`: Runs linting tools to check code quality and style conformity.
- `ng deploy`: Deploys the AngularJS application to a hosting provider.

—---

2. What is jQuery Mobile?

jQuery Mobile is a touch-optimized web framework designed for creating responsive and mobile-friendly websites and web applications. It is built on top of jQuery, a popular JavaScript library, and provides a set of UI components and widgets that are optimized for touch and mobile devices. jQuery Mobile simplifies the process of developing mobile web applications by offering a unified set of tools and controls for building consistent user interfaces across different platforms.

3. What are the benefits (or features) of jQuery Mobile?

jQuery Mobile offers several benefits and features, including:

- **Cross-Platform Compatibility:** jQuery Mobile is compatible with a wide range of mobile devices, tablets, and desktop browsers, ensuring consistent behavior and appearance across different platforms.
- **Responsive Design:** jQuery Mobile provides a responsive grid system and flexible layout options that adapt to various screen sizes and orientations, making it easy to create mobile-friendly interfaces.
- **Touch-Optimized UI Components:** jQuery Mobile offers a rich set of touch-optimized UI components such as buttons, lists, forms, navigation bars, and dialogs, allowing developers to create intuitive and interactive user interfaces for mobile devices.
- **Themeable:** jQuery Mobile comes with a theming framework that allows developers to customize the appearance of their applications using predefined themes or by creating custom themes to match their branding or design requirements.
- **Accessibility:** jQuery Mobile emphasizes accessibility and includes features such as ARIA support and keyboard navigation, making it easier for users with disabilities to navigate and interact with web applications.
- **Performance:** jQuery Mobile is lightweight and optimized for performance, with features such as lazy loading and AJAX navigation to improve page load times and enhance the overall user experience.

4. What is npm?

npm stands for Node Package Manager. It is the default package manager for Node.js, a JavaScript runtime environment. npm is used to install, manage, and share packages of reusable code written in JavaScript. Developers use npm to install packages and dependencies for their projects, publish their own packages, and manage project dependencies efficiently.

5. What are different `data-role` attribute values?

In jQuery Mobile, the `data-role` attribute is used to define the role or behavior of an element in the UI. Some common `data-role` attribute values include:

- `data-role="page"`: Defines a container for a single page in a jQuery Mobile application.
- `data-role="header"`: Defines the header section of a page, typically containing the page title and navigation elements.
- `data-role="content"`: Defines the main content area of a page, where the primary content is displayed.
- `data-role="footer"`: Defines the footer section of a page, typically containing navigation links or copyright information.
- `data-role="button"`: Defines a clickable button element.
- `data-role="listview"`: Defines a list view, which is used to display lists of items.
- `data-role="dialog"`: Defines a dialog box or popup window that overlays the main content.

1. What is AWS VPC?

AWS VPC (Virtual Private Cloud) is a virtual network environment provided by Amazon Web Services (AWS) that allows users to launch AWS resources in a logically isolated section of the AWS cloud.

2. What is AWS Elastic Beanstalk?

AWS Elastic Beanstalk is a fully managed platform-as-a-service (PaaS) offered by AWS that allows developers to deploy and manage web applications and services without worrying about the underlying infrastructure.

3.What are the benefits of deploying a web application on AWS VPC or Elastic Beanstalk?

Deploying a web application on AWS VPC or Elastic Beanstalk offers several benefits, including:

- Scalability: Easily scale resources up or down based on demand.
- High availability: Ensure reliable access to the application with built-in redundancy and failover capabilities.
- Security: Leverage AWS security features to protect the application and data.
- Simplified management: AWS manages infrastructure provisioning, scaling, and monitoring, allowing developers to focus on application development.

4. What are the steps involved in deploying a web application on AWS VPC?

The steps typically involve:

- Creating a VPC and configuring subnets, route tables, and security groups.
- Launching EC2 instances to host the web application.
- Configuring load balancers, auto-scaling groups, and other networking components.
- Deploying the web application code to the EC2 instances.
- Configuring DNS settings to point to the application.

5. What are the steps involved in deploying a web application on AWS Elastic Beanstalk?

The steps typically involve:

- Creating an Elastic Beanstalk environment and selecting the appropriate platform (e.g., Node.js, Java, Python).
- Uploading the web application code or linking to a version control repository.
- Configuring environment settings such as instance type, auto-scaling options, and environment variables.
- Deploying the application to the Elastic Beanstalk environment.
- Monitoring and managing the application using the Elastic Beanstalk console or CLI.

6. What factors should be considered when choosing between AWS VPC and Elastic Beanstalk for deploying a web application?

Factors to consider include:

- Control and customization: VPC offers more control and customization options for networking and infrastructure, while Elastic Beanstalk abstracts away much of the underlying complexity.
- Management overhead: Elastic Beanstalk automates infrastructure management, while VPC requires more manual configuration and management.
- Scalability requirements: Elastic Beanstalk provides built-in auto-scaling capabilities, making it suitable for applications with variable traffic patterns, while VPC requires manual scaling setup.
- Budget: Elastic Beanstalk may be more cost-effective for small to medium-sized applications, while VPC offers more flexibility but may incur higher costs for management and maintenance.