The backbone of any website, HTML (HyperText Markup Language) structures web content Structures web content, making it essential for creating the skeleton of web pages. **HTML** Example: Hello, world! creates a paragraph element. Cascading Style Sheets style the visual presentation of HTML elements Styles HTML elements, crucial for the visual appeal and user experience of web pages. CSS Example: body { font-family: Arial; } applies Arial font to the body. A scripting language that enables interactive web pages. Adds interactivity to web pages, key for dynamic content and responsive user interfaces. **JavaScript** Example: alert('Hello, world!'); displays a message box. A distributed version control system for tracking changes in source code. Facilitates version control, allowing developers to track and manage code changes efficiently. Git Example: git commit -m "Initial commit" commits your changes with a message. A web-based hosting service for version control using Git. Hosts code and facilitates collaboration, making it vital for team-based projects and open-source contributions. GitHub Example: Hosting a repository for your project code. CSS preprocessors that extend CSS with variables, nested rules, and more. Enhance CSS with advanced features, improving code organization and maintenance. SASS/LESS Example (SASS): \$primary-color: #333; body { color: \$primary-color; } A CSS layout model for designing flexible and responsive layout structures. Simplifies layout design, essential for creating flexible and responsive designs efficiently. Flexbox Example: display: flex; creates a flex container. A CSS framework for creating two-dimensional layouts. Enables complex web layouts, crucial for designing grid-based user interfaces with ease. **CSS Grid** Example: display: grid; grid-template-columns: repeat(3, 1fr); creates a three-column grid. A JavaScript library for building user interfaces. Facilitates building user interfaces with reusable components, improving development efficiency and performance. React Example: <Button /> could represent a reusable button component. Provides a way to pass data through the component tree without having to pass props down manually. Streamlines state management in React apps, reducing the complexity of passing data through components. **Context API** Example: Using React.createContext to create a theme context. Application Programming Interface, a set of rules that allow programs to communicate with each Allows different software pieces to communicate, essential for integrating third-party services and data. **API** Example: The Twitter API allows you to fetch tweets. A predictable state container for JavaScript apps, often used with React. Manages application state globally, simplifying state management in complex applications. Redux Example: Storing the state of a user interface in a Redux store. A JavaScript runtime built on Chrome's V8 JavaScript engine for server-side scripting. Enables JavaScript to run server-side, essential for building scalable network applications. NodeJS Example: const http = require('http'); to include the HTTP module. A web application framework for Node.js, designed for building web applications and APIs. Streamlines web application and API development on Node.js, offering a robust set of features for web and mobile applications. **Express** Example: app.get('/', function (req, res) { res.send('Hello World!'); }); A NoSQL database designed to store large amounts of data in document-oriented storage. Provides a flexible, document-oriented database, ideal for applications requiring scalability and flexibility. MongoDB Example: Using db.collection.insertOne() to insert a document into a collection. A powerful, open-source object-relational database system. queries and data integrity. **PostgreSQL** Example: SELECT * FROM users; to select all users from the users table. A delightful JavaScript Testing Framework with a focus on simplicity. Simplifies testing in JavaScript, promoting reliable and bug-free code through easy-to-write tests. Jest Example: test('adds 1 + 2 to equal 3', () => { expect(1 + 2).toBe(3); }); A feature-rich JavaScript test framework running on Node.js and in the browser. Provides a flexible testing framework, essential for ensuring application reliability and performance through comprehensive tests. Mocha Example: describe('Array', function() { it('should start empty', function() { var arr = []; assert.equal(arr.length, 0); }); }); A platform for developing, shipping, and running applications in containers. Facilitates application deployment in containers, ensuring consistency across environments and simplifying deployment processes. Docker Example: Using docker build to create a Docker image of your application. A cloud platform as a service (PaaS) supporting several programming languages. Simplifies application deployment, allowing developers to focus on development without worrying about infrastructure. Heroku Example: Deploying a web app to Heroku with git push heroku master. A cloud platform for static sites and Serverless Functions that fits perfectly with your workflow. Optimizes front-end deployment, ideal for projects prioritizing performance and developer experience. Vercel Example: Deploying Next.js applications with zero configuration. A progressive JavaScript framework for building user interfaces.

FULL STACK WEB DEVELOPMENT Curriculum

HTML | CSS | JavaScript | Git | GitHub | SASS/LESS | Flexbox | CSS Grid | React | Context API | Redux | Node.js | Express | MongoDB | PostgreSQL | Jest | Mocha | Docker | Heroku | Vercel | Vue | Angular | Flask | Django

FULL STACK WEB DEVELOPMENT - KEY TOPICS

Offers a powerful, open-source relational database system, crucial for applications requiring complex Offers a progressive framework for building user interfaces, blending the best of Angular and React's concepts. Vue Example: <template><div>{{ message }}</div></template> in a Vue component. A platform and framework for building single-page client applications using HTML and TypeScript. Provides a comprehensive framework for building scalable applications, including everything from development to testing. Angular Example: <app-root>Loading...</app-root> as the root component. A micro web framework written in Python for building web applications. Offers simplicity and flexibility for web application development, ideal for small to medium projects and prototyping. Flask Example: app.route('/') to define the root URL endpoint. A high-level Python Web framework that encourages rapid development and clean, pragmatic design Facilitates rapid development of secure and maintainable web applications, offering a high-level framework with lots of built-in functionalities. Django Example: Using django-admin startproject mysite to create a new Django project.