A wide range of tasks and abilities are included in web development, such as web design, front-end, back-end, and web server configuration. Here's a more thorough synopsis:

Web design: The visual components of a website, such as the layout, colour palette, and typography, are created by web designers. They establish mockups and prototypes of the website's design using programmes like Adobe Photoshop, Sketch, or Figma.

Front-end Development: Front-end developers construct a website's user-facing interface. Their work involves creating responsive and interactive web pages using languages such as HTML, CSS, and JavaScript. To speed up the development process, front-end frameworks like React, Angular, and Vue.js are frequently used.

Back-end Development: The server-side logic of a website is the responsibility of back-end developers. They speak languages such as To create the back-end code that communicates with the database and responds to user requests, use Python, Ruby, PHP, or Node.js. Back-end developers frequently use frameworks like Django, Ruby on Rails, Laravel, and Express.js.

Front-end and back-end development are two areas in which full-stack developers excel. They are capable of handling every facet of a web application, including server-side logic and database management in addition to UI design.

Configuring Web Servers: In order to host their websites, web developers frequently need to configure web servers. This include managing domain names and DNS settings, configuring security settings, and installing the server software.

Database management: To store and retrieve content for their websites, web developers frequently work with databases. They make use of MongoDB, PostgreSQL, and MySQL database management systems.

The database can be created, read, updated, and deleted using MySQL, PostgreSQL, MongoDB, or SQLite.

Web standards and best practices: To make sure their websites are search engine optimised, responsive, and accessible, web developers must stay current on web standards and best practices. Semantic HTML, CSS for styling and layout, and JavaScript for interactivity are some examples of this.