SECTION 6

TYPES OF SOFTWARE DEVELOPMENT

Desktop Programming

Desktop programming refers to the development of software applications that run on personal computers (PCs), typically in a desktop environment (e.g., Windows, macOS, Linux). These applications are designed for end users to interact with, and they often have graphical user interfaces (GUIs).

Desktop applications can perform a wide range of functions, from word processing and media editing to complex data analysis and gaming. Media editing refers to the process of modifying digital media, such as images, audio, video, and other multimedia elements.

Desktop applications are typically platform-dependent, meaning they are designed for a specific operating system. These programs often use GUIs to allow users to interact with the program. GUI elements like buttons, text fields, drop-down menus, and windows play a critical role in creating a user-friendly experience.

Web Programming

Web programming, also known as web development, refers to the process of creating and maintaining websites or web applications. It involves writing code to build the functionality and design of a site or app that runs in a web browser. Web programming can be divided into two main categories: front-end and back-end development, though full-stack developers handle both.

Front-end development deals with the part of the website or web application that users interact with directly. This includes everything users see and experience, such as layout, navigation, and content presentation.

Front-end developers are responsible for the following:

- Designing the layout and structure of web pages.
- Ensuring websites are responsive, so they function well on desktops, tablets, and other mobile devices.
- Improving user interactivity using JavaScript.

Back-end development involves the server-side of web programming, which is responsible for managing the databases, servers, and the logic that supports the front-end. It handles requests from users and delivers the appropriate content.

Back-end developers are responsible for the following:

- Handling data processing and database interactions.
- Implementing business logic and security features.
- Managing server configurations and ensuring uptime.

Business logic in programming refers to the rules and processes that determine how data is handled and how a system behaves based on specific actions or inputs.

Web programming comes with several challenges. Browser compatibility is one of the key issues, as ensuring that a website functions correctly across different browsers and devices can be difficult. Security is another major concern, as web applications are vulnerable to various attacks such as SQL injection, and data breaches, making it essential to implement robust security measures.

A data breach is an incident where unauthorized individuals gain access to sensitive, confidential, or protected data. This can include personal information, financial records, login credentials, or business secrets. Data breaches can occur due to cyberattacks, system vulnerabilities, or human errors.

Securing Web Applications

In web applications, ensuring that users can securely access their accounts is crucial. When users register (or sign-up) on a site, they create an account by providing personal information, often including login credentials such as a username and password.

After registering, they can log in (or sign-in) by entering their credentials, which are then authenticated by the system. When users wish to end their session, they can log out (or sign-out), which terminates their access to ensure security.

Authentication is the process of verifying the identity of the user by checking their credentials against stored data in a database, ensuring that they are who they claim to be. Common methods of authentication include passwords, multi-factor authentication (MFA), or biometric data like fingerprints.

Authentication does not automatically grant access to all areas of the application. Authorization occurs after successful authentication and determines what specific actions or resources the user is allowed to access based on their role. An unauthorized user, even if authenticated, may be restricted from accessing particular parts of the application.

Mobile Programming

Mobile programming is the process of creating applications (apps) for mobile devices such as smartphones and tablets. These apps can be developed for different operating systems, mainly Android and iOS.

Types of Mobile Applications

There are three main types of mobile apps:

- Native Apps – Built specifically for one platform (Android or iOS). They offer the best performance and full access to device features, including the camera, microphone, location, Bluetooth, and WiFi.

- Web Apps – Mobile-friendly websites that work in a browser (e.g., Progressive Web Apps or PWAs). They do not require installation.
- Hybrid Apps – A mix of native and web apps. They use a single codebase but run on both Android and iOS (e.g., apps built with Flutter or React Native). They are downloaded from an app store but mainly run web code.

Embedded Programming

Embedded programming refers to the process of writing software for embedded systems—specialized computing devices designed to perform specific tasks. Unlike general-purpose computers (such as desktops or laptops), embedded systems are optimized to execute one or a few specific functions reliably and efficiently. These systems typically have constraints on resources such as memory, processing power, storage, and input/output (I/O) options.

Applications of embedded programming are widespread and include devices such as smart TVs, microwave ovens, washing machines, coffee machines, digital cameras, smart thermostats, and medical devices.