## horizontal line



Project Name

09.04.20XX

**─**

Your Name

Your Company

123 Your Street

Your City, ST 12345

# Report on Web Programming Concepts

## 1. Authentication

Authentication is basically checking if the user is who they say they are. It’s like a digital bouncer making sure you have the right username and password to get in. Sometimes, it’ll even check your device or where you are, depending on how sensitive the data is.

Example: When you log into your email, you punch in your username and password, and the system gives it a once-over to make sure you’re not an imposter.

## 2. Web Security

Web security is all about putting up walls against cybercriminals who want to steal data or invade your privacy. Developers do this by using tools and techniques to keep the bad guys out.

Example: One easy move? Using HTTPS instead of HTTP to add a layer of security to the website.

## 3. Hypertext Transfer Protocol (HTTP)

HTTP is the basic language that lets data travel between a server and a web browser. Every website has a server, and when you type in a URL, HTTP is what makes your request for the website’s data possible.

Example: Think of it like this—when you want to visit a site, you enter its address in your browser. HTTP is what delivers the website from the server to your screen.

## 4. Session Management

Session management is how a web app keeps track of what you’re doing as you click around. It remembers your preferences and actions, so you don’t have to log in every time you switch pages. This usually involves creating a session ID that gets saved and used to keep everything consistent as you browse.

Example: After you log into a site, session management lets you move from page to page without needing to sign in again. It’s like your digital breadcrumb trail.

## 

## 

## 5. Principles of Stateless Programming

Stateless programming is a fancy way of saying that each request to a server stands alone, with no memory of what happened before. The server doesn’t store any past interactions, so each request needs all the info to be processed on its own. This approach makes servers simpler and easier to scale since there’s no need to keep track of past requests.

Example: RESTful web services work this way. Every API call has everything it needs to complete the request, and the server doesn’t remember what you asked for five minutes ago.