

# Web Applications

Garcia Gonzalez Christian Andres

January 2024

## I. INTRODUCTION

A web application is software that runs in your web browser. Businesses have to exchange information and deliver services remotely. They use web applications to connect with customers conveniently and securely. The most common website features like shopping carts, product search and filtering, instant messaging, and social media newsfeeds are web applications in their design. They allow you to access complex functionality without installing or configuring software.

## II. BENEFITS OF WEB APPLICATIONS

Web applications have several benefits, with almost all major enterprises utilizing them as part of their user offerings. Here are some of the most common benefits associated with web apps.

### A. Accessibility

Web apps can be accessed from all web browsers and across various personal and business devices. Teams in different locations can access shared documents, content management systems, and other business services through subscription-based web applications.

### B. Efficient development

As detailed, the development process for web apps is relatively simple and cost-effective for businesses. Small teams can achieve short development cycles, making web applications an efficient and affordable method of building computer programs. In addition, because the same version works across all modern browsers and devices, you won't have to create several different iterations for multiple platforms.

### C. User simplicity

Web apps don't require users to download them, making them easy to access while eliminating the need for end-user maintenance and hard drive capacity. Web applications automatically receive software and security updates, meaning they are always up to date and less at risk of security breaches.

### D. Scalability

Businesses using web apps can add users as and when they need, without additional infrastructure or costly hardware. In addition, the vast majority of web application data is stored in the cloud, meaning your business won't have to invest in additional storage capacity to run web apps.

## III. COMMON WEB APPLICATIONS

There are numerous types of web applications. Here are some of the most well-known.

### A. Workplace collaboration web applications

Workplace collaboration web apps allow team members to access documents, shared calendars, business instant messaging services, and other enterprise tools.

### B. Ecommerce web applications

Ecommerce web apps such as Amazon.com enable users to browse, search, and pay for products online.

### C. Email web applications

Webmail apps are widely used by enterprises and personal users to access their emails. They often include other communication tools such as instant messaging and video meetings.

### D. Online banking web applications

Business and personal users widely use online banking web apps to access their accounts and other financial products such as loans and mortgages.

## IV. HOW DO WEB APPLICATIONS WORK?

Web applications have a client-server architecture. Their code is divided into two components—client-side scripts and server-side scripts.

### A. Client-side architecture

The client-side script deals with user interface functionality like buttons and drop-down boxes. When the end user clicks on the web app link, the web browser loads the client-side script and renders the graphic elements and text for user interaction. For example, the user can read content, watch videos, or fill out details on a contact form. Actions like clicking the submit button go to the server as a client request.

### B. Server-side architecture

The server-side script deals with data processing. The web application server processes the client requests and sends back a response. The requests are usually for more data or to edit or save new data. For example, if the user clicks on the Read More button, the web application server will send content back to the user. If the user clicks the Submit button, the application server will save the user data in the database. In some cases, the server completes the data request and sends the complete HTML page back to the client. This is called server side rendering.

## V. COMMON WEB APPLICATION SECURITY RISKS

Web applications may face various attack types based on the attacker's goals, the nature of the targeted organization's work, and the application's security gaps. Common attack types include:

- **Zero-day vulnerabilities:** These are vulnerabilities unknown to an application's makers, often exploited before fixes are available.
- **Cross-site scripting (XSS):** Allows attackers to inject client-side scripts, compromising user information.
- **SQL injection (SQI):** Exploits vulnerabilities in database query execution, gaining unauthorized access to information.
- **Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks:** Overload servers with attack traffic, causing service denial.
- **Memory corruption:** Unintentional modification of memory, exploited through code injections or buffer overflow attacks.
- **Buffer overflow:** Anomaly where software writing data overflows a defined memory buffer, potentially creating vulnerabilities.
- **Cross-site request forgery (CSRF):** Tricking users into making unauthorized requests, often targeting privileged accounts.
- **Credential stuffing:** Use of bots to input stolen username/password combinations into login portals.
- **Page scraping:** Bots stealing content from webpages for various purposes.
- **API abuse:** Exploiting vulnerabilities in Application Programming Interfaces (APIs) to send malicious code or intercept data.
- **Shadow APIs:** APIs built without informing security teams, exposing sensitive data.
- **Third-party code abuse:** Exploiting vulnerabilities in third-party tools integrated into web applications.
- **Attack surface misconfigurations:** Vulnerabilities due to overlooked or misconfigured elements in the organization's attack surface.

## VI. WEB APPLICATION SECURITY STRATEGIES

Web application security is a dynamic discipline, and best practices evolve with emerging threats. However, certain fundamental security services are crucial:

- **DDoS Mitigation:** Prevents overwhelming servers with malicious traffic.
- **Web Application Firewall (WAF):** Filters out traffic exploiting web application vulnerabilities.
- **API Gateways:** Identifies and manages APIs, blocking traffic targeting API vulnerabilities.
- **DNSSEC:** Ensures safe routing of DNS traffic to correct servers.
- **Encryption Certificate Management:** Third-party management of SSL/TLS encryption elements.
- **Bot Management:** Uses machine learning to distinguish automated traffic from human users.

- **Client-side Security:** Checks for new third-party JavaScript dependencies and code changes.
- **Attack Surface Management:** Provides tools to map and mitigate risks in the attack surface.

## VII. WHAT ARE NATIVE APPS?

A native mobile app has been programmed to run on a specific device and written in a platform-definite programming language. For instance, Objective-C and Swift languages are used to build native apps in iOS.

On the other hand, Kotlin and Java are the preferred programming languages for native Android applications. Whatever platform you choose for native app development, you will have dedicated development tools, SDK, and interface elements for each platform.

Since native mobile app development utilizes platform-specific programming language, you cannot build an application that will run on both platforms, i.e., iOS and Android. That means an app built for iOS will not exist for download at the Google Play Store or work on an Android device.

The same applies to native applications built for Android. It will not be available in the App Store or cannot be used on an iPhone device. There are several native apps examples, including WhatsApp, Spotify, Pokémon Go, etc.

### A. Pros and Cons of Native Apps

Here are some key advantages of native apps:

- Native applications are customized for particular platforms, and therefore, they are highly responsive.
- Native apps are high-performance and feature high speed that helps enhance the user experience.
- As they are available for download in app stores, you can potentially reach a broader audience base who are active users of mobile applications.
- Every mobile platform has its standard UI practices, and native app developers need to follow these to deliver an improved experience.

#### 1) Disadvantages of Native Apps::

- Native apps are built using complex programming languages, which require a proficient developer with experience in building these types of applications.
- Native app development is costlier than web or hybrid apps. This is because separate applications need to be built, deployed, and maintained for each mobile platform if you want to maximize your reach.
- If you are planning to develop a simple application for your business or users, native apps may not be the best choice.
- Native web apps do not support cross-platform compatibility. An application built for iOS will not work on an Android device.

## VIII. WHAT ARE HYBRID APPS?

When the best of web and native comes together, you get a hybrid mobile app. They render the look and feel of a web app but can be downloaded and installed like a native app.

Simply put, hybrid applications behave like native apps and work across platforms like web apps. These applications are built with CSS, JavaScript, or HTML and run in a web view.

On the one hand, hybrid apps are platform-agnostic; on the other hand, they are multi-platform compatible, i.e., the same codebase can be distributed across diverse platforms. These are the perfect combination of both worlds and are developed utilizing the best hybrid application development frameworks. They harness the benefits of native while allowing the same app's availability on both iOS and Android platforms.

Thus, businesses can choose web application development outsourcing services and maximize their reach through hybrid app development by making the app available on the leading platforms. Additionally, developers can access dedicated development tools, SDK, and interface elements like native apps. There are significant hybrid apps examples, including Instagram, Uber, Gmail, and Evernote.

#### A. Pros and Cons of Hybrid Applications

Let's now go through the advantages of hybrid app development:

- Hybrid applications are faster to develop because a significant part of the development process is done through standard web technologies and efficient hybrid application frameworks.
- Developers need to build only one codebase that can be deployed across multiple platforms. No need to write unique codes for each platform. This also helps save time in the hybrid app development process.
- You can reach a wide audience base by building cost-effective hybrid apps that can be distributed on both App Store and Play Store.
- Building and deploying updates for hybrid web apps is more straightforward. The applications are also easy to maintain with minimal glitches.

##### 1) Cons of Hybrid Apps::

- Hybrid apps are comparatively slower and fail to deliver some native-like experiences. This is because these apps are developed simultaneously for two platforms.
- Hybrid app development is costlier than web apps because they depend on a third-party system and need a wrapper.
- Hybrid web applications are less intuitive and interactive than native applications and, therefore, are not suitable if you want to build highly interactive apps for improved user experiences.
- If you are trying to make customizations, the core advantage of the hybrid may be compromised.

#### REFERENCES

- [1] Patadiya, J. (2023, 21 de diciembre). *Web vs. native vs. hybrid apps: Which is better?* Recuperado de <https://radixweb.com/blog/web-vs-native-vs-hybrid-application>
- [2] *What is web application security?* — *Web security* — Cloudflare. (s.f.). Cloudflare. Recuperado de <https://www.cloudflare.com/learning/security/what-is-web-application-security/>

- [3] *What is a Web App? - Web Application Explained* - AWS. (s.f.). Amazon Web Services, Inc. Recuperado de <https://aws.amazon.com/what-is/web-application/>