Session: 18

HTML5 Web Storage

Objectives

- Explain Web storage in HTML5
- Explain session storage
- Explain local storage
- Explain the Indexed DB API
- Describe a native app
- Explain the difference between native apps and HTML5 apps
- Describe the advantages of native and HTML5 apps
- List the steps to convert HTML5 apps to native apps

Introduction

- Traditionally, Web applications use cookies to store small amounts of information on a user's computer.
- A cookie is a file that stores user-related information, such as login details, and may either be temporary or permanent.
- Drawbacks of cookies :
  - slow down the performance of Web application, as they are included with every HTTP request
  - cannot be considered as safe means for transmission of sensitive data
  - cannot store large amount of information, as they have a limitation of size of 4 KB
- W3C has designed a specification named Web Storage API which offer a solution to store data on the client-side

Web Storage in HTML5

- Is a W3C specification and certain browsers refer to it as 'DOM Storage'.
- Provides functionality for storage of data on the client-side that is on user's machine.
- Stores data that can cater for both temporary as well as permanent needs.
- Offers more control than traditional cookies, and is easy to work with.
- Was originally a part of the HTML5 specification, but now it is present in a separate specification and stores a maximum of 5 MB of information per domain.

Browser-specific Web Storage

- Web storage is browser-specific and the location where the Web storage data is stored depends on the browser.
- Each browser's storage is separate and independent, even if it is present on the same machine.
- HTML5 web storage is implemented natively in most browsers, so one can use it even when third-party plug-in is not available.
- List of browsers support for HTML5 Web storage:

Browser	Version
IE	8.0+
Firefox	3.6+
Safari	4.0+
Chrome	5.0+
Opera	10.5+

Session Storage

- To assign data:

```
sessionStorage.setItem(key, value);
```

where, key: is the named key to refer data

value: is the data to be stored

- To retrieve data:

```
var item = sessionStorage.setItem(key);
```

- To remove data:

```
sessionStorage.removeItem(key);
```

- To clear data:

```
sessionStorage.clear();
```

Local Storage

- Enables to save data for longer periods on the user's computer
- Data is persistent and can be retrieved when a user visits the site again.
- Is used, if data needs to be stored for more than a single session.
- Works in a similar fashion as session storage.
- Uses the same functions, such as **setItem(), getItem(), removeItem(),** and **clear().**

Indexed Database API 1-3

- A database is an organized collection of data. Example, a relational database stores the data in the form of tables.
- A table comprises rows and columns that are used to store data.
- The representation of data from a table is in the form of records.
- HTML5 has introduced a new Web Storage API which can host Web databases locally within the user browser.
- Web databases are not like relational databases in terms of functionality.

Indexed Database API 2-3

- Indexed Database API is also known as IndexedDB.
- It is an object store that can be used to store and manipulate data on the client-side, within the browser.
- IndexedDB enables to create Web applications with rich query abilities and which can work both online and offline.
- IndexedDB supports two types of API namely, synchronous and asynchronous.
- The synchronous API can be used with WebWorkers, whereas asynchronous API can be used for Web applications.

Indexed Database API 3-3

- IndexedDB API is implemented using **window.indexedDB** object.
- Browsers implement the IndexedDB object with their own prefixes. For example, Chrome uses the **webkit**, whereas Mozilla supports **–moz** prefix.
- Following table lists the browser support for the IndexedDB API.

IE	Firefox	Chrome	Safari	Opera	iOS Safari
6.0	-	-	-	-	3.2
7.0	8.0moz	-	-	-	4.0-4.1
8.0	9.0moz	16.0webkit	5.0	-	4.2-4.3
9.0	10.0moz	17.0webkit	5.1	11.6	5.0
10.0	11.0moz	18.0webkit	6.0	12.0	-
-	12.0moz	19.0webkit	-	-	-

Implementing IndexedDB API

Steps to implement the IndexedDB API in a Web app are as follows:

1. Open a database
2. Create an object store
3. Start a transaction
4. Perform some database operations, such as add and retrieve
5. Work with the retrieved results

Limitations of IndexedDB API

- Internationalized sorting deals with sorting of string data. As the database does not follow any international order for storing data, internationalized sorting is not supported by the API.
- IndexedDB API does not synchronize client-side database with the server-side databases.
- IndexedDB API supports querying the client-side database, but does not support the use of operators, such as LIKE that is used by Structured Query Language (SQL).

Native Apps

- A native application also known as native app is an application program that is built for using it on a particular device or platform.
- A native app, when compared with Web app, is installed on a device and has a faster response, because it has a direct user interface.
- BlackBerry Messenger (BBM) is a native app available on blackberry mobile devices.

Native Apps vs HTML5 Apps

- HTML5 web apps are accessible and used on any devices through Web browser similar to the mobile Web site.
- HTML5 Web apps have the ability of offline access which means that the user need not have a network connection.

Native Apps	HTML5 Apps
runs on iOS and Android devices that can be downloaded or purchased from the online app stores.	runs on a Web server, usually in a Web browser.
use programming language, such as Java for Android devices and Objective C for iOS devices.	Web developers use HTML, JavaScript, and CSS.

Advantages of HTML5 Apps

- **Users cannot identify the differences** - cannot identify whether they are working on a hybrid HTML5-native app or a fully native app or an HTML5 app.
- **Users adjust styles for devices** - HTML5 apps can be viewed on any devices that contains Web browser.
- **Upcoming functionalities** - HTML5 does not support all features on a device, but it is coming up with new functionalities.
- **Improving Performance** - Many developers learn new methods to improve the performance of Web.
- **Independent device** - If the developers want that their app to be used by a large number of users, they should design and develop apps for both mobile users as well as desktop users.
- **Developers are not locked in app stores** - HTML5 developers are not restricted to an app store. Instead, they can create applications and sell them like any other Web page.

Advantages of Native Apps

Major advantage of native apps over HTML5 apps is that they are faster than HTML5 apps. Native apps provide more benefits over HTML5 apps as follows :

- **Providing access to device hardware** - There are no APIs available for accelerometers, cameras, or any other device hardware for HTML5 apps.
- **Uploading Files** - Native apps can access the file system in Android and some files in iOS. However, the HTML apps cannot.
- **Push Notifications** - The push notifications are sent always with an open IP connection to apps on the iOS device.
- **Accessing device files** - Native apps communicate with files on devices, such as contacts and photos. However, HTML5 app cannot.
- **Superior graphics than HTML5** - HTML5 has a canvas element, but it will not create a full 3D experience.
- **Offline access** - HTML5 provides access to offline Web applications. However, a native app is stored on local machine, so the users does not require access to the Web to work with the application.

Converting HTML5 Apps to Native Apps

- Users have a choice of developing their application in HTML5 and convert them into a native app
- They can use some tools to convert an HTML5 app to Native app and they are as follows:
  - PhoneGap - Is an HTML5 app that allows the user to create native apps with Web technologies and is accessible to app stores and APIs.
  - Appcelerator - Is a cross-platform mobile application development support and allows the users to create Android, iOS, and mobile Web apps.

Summary

- Web Storage is a W3C specification that provides functionality for storing data on the client-side for both temporary as well as permanent needs.
- HTML5 Web applications make use of Web storage to implement client-side persistent storage and they are: session storage and local storage.
- Session storage keeps track of data specific to one window or tab.
- The setItem() and getItem() methods are used to store and retrieve the data from session storage.
- Local storage enables to save data for longer periods on the user's computer, through the browser.
- IndexedDB API is basically an object store that can be used to store and manipulate data on the client-side.
- A native application also called as native app is an application program that is built for a particular device or platform.